May 1980
$2.50

kilobaud

MICROCOMPUTING T.M.

for business . . . education . . . FUN!

"On the Air"

A radio station computerizes.
P. 22

Two H8 Projects:

Timer for the darkroom. **P. 34**
Expansion chassis. **P. 40**

Inexpensive Interconnection

Going multi-user on a budget.
P. 104

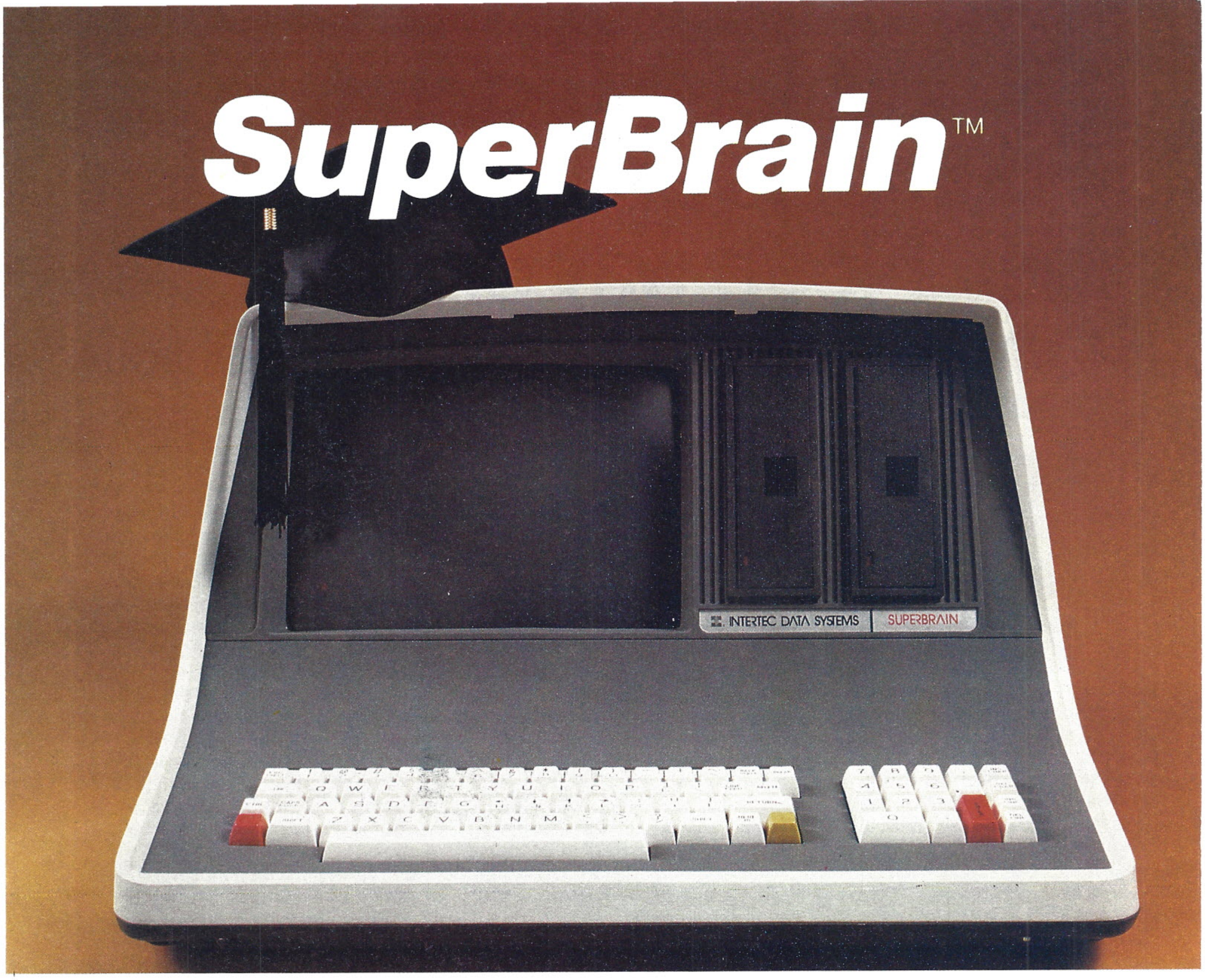
JazZ-80

Do it in 8-part harmony. **P. 148**

Kid Power

Never too old—nor too young—to learn. **P. 204**

05
0 74470 65945



The Honor Graduate

There's been a lot of talk lately about intelligent terminals with small systems capability. And, it's always the same. The systems which make the grade in performance usually flunk the test in price. At least that was the case until the SuperBrain graduated with the highest PPR (Price/Performance Ratio) in the history of the industry.

For less than $3,000*, SuperBrain users get exceptional performance for just a fraction of what they'd expect to pay. Standard features include: two dual-density mini-floppies with 320K bytes of disk storage, up to 64K of RAM to handle even the most sophisticated programs, a CP/M Disk Operating System with a high-powered text editor, assembler and debugger. And, with SuperBrain's S-100 bus adapter, you can even add a 10 megabyte disk!

More than an intelligent terminal, the SuperBrain outperforms many other systems costing three to five times as much. Endowed with a hefty amount of available software (BASIC, FORTRAN, COBOL), the SuperBrain is ready to take on your toughest assignment. You name it! General Ledger, Accounts Receivable, Payroll, Inventory or Word Processing . . . the SuperBrain handles all of them with ease.

Your operators will praise the SuperBrain's good looks. A full ASCII keyboard with a numeric keypad and function keys. A non-glare, dynamically focused, twelve inch screen. All in an attractive desktop unit weighing less than a standard office typewriter. Sophisticated users will acclaim SuperBrain's twin Z-80 processors which transfer data to the screen at 38 kilobaud! Interfacing a printer or modem is no problem using SuperBrain's RS-232C communications port. But best of all, you won't need a PhD in computer repair to maintain the SuperBrain. Its single board design makes servicing a snap!

So don't be fooled by all the freshman students in the small systems business. Insist on this year's honor graduate . . . the SuperBrain.

INTERTEC DATA SYSTEMS®

2300 Broad River Road, Columbia, SC 29210
(803) 798-9100 TWX: 810-666-2115

*Quantity one. Dealer inquiries invited.

kilobaud
MICROCOMPUTING T.M.

contents: may '80

ARTICLES

DEPARTMENTS

Cover: Program Director Tim Tobin on the air at WSCV/WSLE in Peterborough NH. Photo by Reese Fowler.

micro info

$ This symbol next to a title in the table of contents indicates that the article is a business-application article.

Manuscripts

Contributions in the form of manuscripts with drawings and/or photographs are welcome and will be considered for possible publication. We can assume no responsibility for loss or damage to any material. Please enclose a self-addressed, stamped envelope with each submission. Payment for the use of any unsolicited material will be made upon acceptance. All contributions should be directed to the *Microcomputing* editorial offices. "How to Write for Microcomputing" guidelines are available upon request.

Editorial Offices:

Pine Street
Peterborough NH 03458
Phone: 603-924-3873

Advertising Offices:

Elm Street
Peterborough NH 03458
Phone: 603-924-7138

Circulation Offices:

Elm Street
Peterborough NH 03458
Phone: 603-924-7296

To subscribe, renew or change an address:

Write to *Microcomputing*, Subscription Department, PO Box 997, Farmingdale NY 11737. For renewals and changes of address, include the address label from your most recent issue of *Microcomputing*. For gift subscriptions, include your name and address as well as those of gift recipients. Postmaster: Send form #3579 to *Microcomputing*, Subscription Services, PO Box 997, Farmingdale NY 11737.

Subscription problem or question:

Write to *Microcomputing*, Subscription Department, PO Box 997, Farmingdale NY 11737. Please include an address label.

Kilobaud *Microcomputing* (ISSN 0192-4575) is published monthly by 1001001, Inc., 80 Pine St., Peterborough NH 03458. Subscription rates in U.S. are $18 for one year and $45 for three years. In Canada: $20 for one year and $51 for three years. In Europe, send 89,-DM in Eurocheque or send credit card information to: Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. South African Distributor: KB Microcomputing, PO Box 782815, Sandton, South Africa 2146. All other foreign subscriptions are $23—one year only (surface mail). Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Phone: 603-924-3873.

Wayne Green

PUBLISHER'S REMARKS

Primal Scream

Back in the July 1979 issue, I wrote about some of the problems we'd been having with our Prime 300 minicomputer. We invested in the system so we would have in-house computing to handle the subscribers' labels and renewals, payroll, bookkeeping, reader service requests, and keep track of potential advertisers, potential subscribers and the article inventory. Now it looks as though the approximately $250,000 we spent for the system has been a complete waste.

Despite my writing to Prime several times, calling them and even mentioning the problems in the magazine, Prime has been unresponsive. I talked with Prime representatives about this at NCC, and though they promised action, nothing has happened yet. I am getting the impression that once they sell a system they care less what happens to the customer.

In the July editorial I reported that we got about 20 percent of what I was led to expect out of the Prime. We gave up on using it for subscriptions, payroll, bookkeeping, article records, ad records and potential subscriber records. We were using it for reader service requests, but those are so far behind as a result of the Prime's breaking down repeatedly that we have moved that function to a service bureau.

About the only job left for it was keeping a mailing list for our "Microcomputing Industry Newsletter." When that simple chore held up the mailing of this publication for three weeks, we moved it to a TRS-80 and had little more for the Prime to do. We had also been trying to handle the daily orders for books and Instant Software, but that, too, got a month behind and was moved to a Midwest Scientific microcomputer.

These days we are doing most of our inhouse computing on either the Midwest system or our TRS-80 systems. The service contract on the Prime is over $750 per month, and you can buy a lot of TRS-80s in a year for that. We have fewer breakdowns with the TRS-80 systems than with the Prime.

I don't know what experience others are having with the Prime. Someone must be happy because Prime keeps building new buildings and growing. Sales in 1979 were over $150 million, and the growth rate was over 50 percent in the last year. Surely all of Prime's customers can't be as disappointed as we have been with the performance we expected from the Prime.

The original idea, as discussed with Prime, was to buy the system they said we needed and have it programmed for publishing-house use. We then intended to sell Prime systems, OEM, to other publishers, figuring that we might be able to sell and program 50 systems or more a year. Now we're looking around for a way to dump our Prime 300 because it can't handle the work required even by a relatively small publishing house.

Job Opportunities

Two nearby colleges are still looking for computerists with formal educational backgrounds for their microcomputer schools. Contact Dean Coles, Franklin Pierce College, Rindge NH 03461, or Dean Jones, Nathaniel Hawthorne College, Antrim NH 03440. Here's an opportunity for you to help set up a computer department and train the people who will become the future chief executives in the growing microcomputer field.

Positions are open in our publishing and software departments. We need book editors, editors for both *Kilobaud* and "80" magazines, marketing people, sales-oriented people, hams to help test new equipment and hardware and software people for Instant Software. We're looking for people who have leadership qualities to move up to middle and top management.

Send a letter telling us what you have done, what you can do for us and what we can do for you to Al Thulander, *Microcomputing*, Peterborough NH 03458. Not only will we expand the magazine and book-department staffs, we plan to publish three new magazines, all of which will need full staffs.

Mailing Schedule Changed

The mailing schedule for *Kilobaud Microcomputing* has been changed a few days to bring the cover date more in line with the date of receipt of the magazine. The publication of three magazines a month forced us to shift around our production, printing and mailing schedules so that one magazine could go to press every ten days. *73* goes to press first each month, followed by *Kilobaud Microcomputing* and *80 Microcomputing*. With 150- to 250-page magazines coming out every ten days, everyone is kept busy.

Sherry Smythe

OUTPUT FROM ISI

Instant Software has released its first disk programs and sent samples to the ISI reps around the world. Most programs will continue to be released on cassettes, though, even if they will be used with disk systems. This makes sense when you consider that cassettes are less expensive than disks.

Cassette tapes are less likely to be damaged in shipment or on display than disks. They are less likely to be accidentally influenced by stray magnetic fields, and they take up less space. Disk packages make sense when the price of the program is high enough so the cost difference between cassettes and disks is a small matter.

ISI disk programs are packaged in a three-ring binder and generally have more documentation than the cassette packages. The first disk release is an energy audit program, which should interest realtors since laws that could prohibit the sale of a building without such an audit are under consideration. This package will also be useful for energy-conservation consultants.

Asian Trip

For about $2000 you can sign up for the coming IEEE tour of consumer electronics shows in Japan, Taiwan, Korea and Hong Kong. This trip will start about the first of October and be back in three weeks. It takes you to four electronics shows where you will be able to meet businessmen interested in importing your products, and the prices are right. If you want to go, please drop me a line and I'll get you full information. Wayne is working on a special trip to add short visits to Macao and even China, if possible.

Robert W. Baker

PET-POURRI

Tape Drive with a Counter

Someone finally did it. There is now a cassette drive with a counter available for the PET at a reasonable price. The D&R S2545 cassette system is supplied by D&R Creative Systems, PO Box 402B, St. Clair Shores MI 48080, and sells for $83 plus $3 shipping and handling. The cassette system consists of a self-contained, epoxy-potted interface module and a Sanyo M2545A recorder.

The module plugs into the cassette port at the rear of the PET and fits flat against the back panel. Because of the module's size, it cannot be used with the cassette interface inside the PET. The recorder is specially modified to sense when the recorder buttons are pressed and operates identically to the PET cassette drives with full program control. Even with the slight modification, the cassette recorder is still fully warranted by Sanyo.

Four cables from the interface module connect to the recorder REM, EAR, MIC and SENSE connectors; the latter is mounted inside the unused battery compartment. In order to connect the SENSE line connector, you must remove the battery compartment cover from the Sanyo recorder.

The D&R cassette system provides two methods of program location. Using the digital counter, programs can index and quickly relocate with their beginning and ending counter values. Also, the D&R system can locate programs audibly using the fast-forward cuing features of the Sanyo recorder. While the recorder is in the Play mode, depress the Fast Forward button and you can hear the programs pass by the recording head at the fast-forward cuing speed. You can clearly detect the 10-second leader at the start of each program.

In testing a D&R cassette system on my PETs, I found no compatibility problems when switching tapes between the D&R cassette and three different Commodore C2N drives.But D&R does warn of possible incompatibilities due to head alignments of the Commodore drives since Commodore changed its cassette unit three times after the PET was introduced. I also found no problems with special settings of the volume and tone controls—you merely turn both clockwise all the way and let the D&R interface do its job.

The only disadvantage of the D&R cassette system is that you can't connect the MIC and EAR cables to the recorder at the same time. Whenever you switch between reading or writing a tape, you must switch the cables to the Sanyo recorder. This slight inconvenience is a small price to pay when you consider the digital counter and fast-forward cuing features provided. Also, you can still use the Sanyo recorder as a standard recorder with built-in microphone and automatic level control.

Programming Tips

Ever save a machine-language program on tape and forget what memory locations it uses? Try this simple trick on systems with the new ROMs. Place the tape in the cassette drive and type OPEN 1 instead of LOAD. This will read the file header but will not load the program into memory. Now type SYS 64785 to enter the machine-language monitor, followed by M 027A,0290 to display the first 24 bytes of the tape buffer. The format of the data found in the buffer is shown in Example 1.

Locations 027B–027E indicate the starting and ending addresses for any program file. Remember that the addresses are in 6502 address format, low byte first followed by the high byte. All values are in hexadecimal.

Here's a simple programming trick for anyone with a printer to allow selectable output for either the printer or the display. Instead of having separate print statements for both the printer and the display, try this. After deter-

Byte 1	027A	File type (01 = program)
Bytes 2/3	027B–7C	Program starting address
Bytes 4/5	027D–7E	Program ending address
Bytes 6+	027F–	File name

Example 1.

OPEN 5,4 opens a logical file (#5) for output to the printer (device #4)
OPEN 5,3 opens a logical file (#5) for output to the display (device #3)

Now simply code all PRINT statements that are to be selectable as printed or displayed as:

PRINT#5,"........ text ...."

and all output will automatically switch as selected without any additional print statements, flag testing, etc. Any normal PRINT statements will still always appear on the display screen as usual.

Example 2.

kilobaud MICROCOMPUTING T.M.

PUBLISHER/EDITOR
Wayne Green

ASSISTANT PUBLISHER/EDITOR
Jeffrey DeTray

MANAGING EDITOR
John Barry

EDITORIAL ASSISTANTS
Dennis Brisson
Susan Gross

ASSOCIATE EDITORS
Frank Derfler, Jr.
Rod Hallen
Peter Stark
Sherm Wantz

PRODUCTION DEPARTMENT
MANAGER:
Noel Self

ASSISTANT MANAGER:
Robin Sloan

STAFF:
Steve Baldwin
James Butler
Robert Drew
Bruce Hedin
Kenneth Jackson
Clare McCarthy
Michael Murphy
Dion Owens
Patrice Scribner
Susan Symonds
John White

PHOTOGRAPHY
William Heydolph
Tedd Cluff
Terrie Anderson
Reese Fowler

TYPESETTING
Barbara Latti
Sara Bedell
Sandie Gunseth
Mary Kinzel
Karen Podzycki

PROJECTS EDITOR
Jim Perry

BOOK EDITORS
Chris Brown
Emily Gibbs

EXECUTIVE VICE PRESIDENT
Sherry Smythe

CORPORATE CONTROLLER
Alan Thulander

EXECUTIVE ASSISTANT
Leatrice O'Neal

ACCOUNTING MANAGER
Knud Keller

CIRCULATION MANAGER
Debra Boudrieau

CIRCULATION
Barbara Block
Pauline Johnstone

BULK SALES MANAGER
Ginnie Boudrieau

EUROPEAN MARKETING DIR.
Reinhard Nedela

ADVERTISING
603-924-7138
Aline Coutu, Mgr.
Marcia Stone
Penny Brooks
Nancy Ciampa
Louise Holdsworth
Jerry Merrifield
Rita Rivard
Kevin Rushalko
Hal Stephens
Phoebe Taylor

mining whether the user wants printed or display output, open a logical file for the appropriate device (see Example 2).

Creative Software Products

Creative Software, PO Box 4030, Mountain View CA 94040, has programs available for the PET at reasonable prices. They include Household Finance and Utility packages as well as various joystick-interface games. Most of these programs will run on any model PET, but check each program to make sure it will run on your system. All the programs are well written and provide an excellent user interface to help people unfamiliar with the programs.

The Household Finance package sells for $20 and consists of two programs designed to record and analyze household expenditures and income. Part I inputs and lists budget items while creating a record on cassette tape of all items entered. Part II sums the data tapes created using part I and presents overall patterns of spending.

Part I allows inputting data for a new month, or updating and adding data to a previously recorded month. You enter the date, amount, category code, payment method (or check number) and any brief description for each item to be recorded. Item descriptions are limited to 29 characters and cannot contain commas. To classify each item, sixteen category codes are available: Auto, Mortgage, Clothes, Entertainment, Education, Food, Gifts, Household, Income, Taxes, Medical, Insurance, Savings, Utilities, Vacation, Miscellaneous. If you have over 100 items in a month or run out of memory, you can use more than one tape for any month.

Part II then reads the data files created by part I and prints a table of the totals for each category. A nice feature of this program is a graphic display of a spending profile that shows what percentage was spent in each category. Other options allow displaying various category totals or adding additional data from other months.

The Household Utility packages are $15 each for three programs on a tape. The first package includes a program to help compare the tradeoffs between buying a house and renting an apartment. It takes into account such items as inflation, property taxes and tax deductions. Another program on this tape computes various loan calculations to find monthly payments, length of a loan and how much you can afford to borrow with a monthly payment limit.

The second Household Utility package is more useful. It includes a Compound Interest program to solve savings-account investment problems concerning the future value of an account. An Amortization program calculates the usual costs associated with a mortgage loan. It displays the current amount paid to interest and principal as well as the total amount paid in interest and the remaining balance on the loan. The last program calculates the cost per mile of driving an automobile. The program takes into account the purchase price of the car, licensing fees, routine maintenance and repairs, insurance, number of miles driven, gas expenses and the approximate resale value.

The Household Finance package can be valuable in establishing and maintaining an effective home budget. However, the chore of recording every expenditure and income may be overburdening if you use tape data files. If you have disk drives available you can modify the program to use disk data files instead of tape. This enhances the excellent features already provided by these programs while drastically reducing the time to read or write data files.

In the January issue I mentioned the joystick interfaces available from Creative Software. The dual joystick interface connects two Atari joysticks and sells for $45 with two demo programs and documentation. The joysticks are not included and cost an additional $15 each if you don't already have an Atari video game.

The interface consists of a sealed, $1 \times 2 \times 3$ inch plastic box. A user port connector is mounted on one side and two subminiature D connectors for the joysticks are mounted on the opposite side. The interface plugs into the PET user port, and either or both joysticks are plugged into the appropriate connectors.

The interface is compatible with all *Cursor* magazine programs that use joysticks for control. However, the user port connector is not "keyed," so you must plug it in right side up. Be careful not to exert any downward pressure on the interface box to avoid damaging the PET main logic board.

Since most games using joysticks also provide sound effects, I was surprised not to find any connector provided for sound output. You can remedy this by adding a miniature phone connector wired to the CB2 and ground lines of the user port connector. There is plenty of room in the interface box for adding this connector and a dropping resistor or miniature switch.

A Fairchild joystick sells for $35 and comes wired directly to a user port connector ready to use. The connector is not keyed, but the top side is identified by a label. The wiring connections are protected on the back of the connector. Multiple actions are possible with the Fairchild joystick.

Both joystick interfaces come with a simple write-up describing their installation and programming. Additional programs are available that allow using either interface or the numeric keypad for motion control. They sell for $10 each and include:

Breakout—the standard game with paddle-size options.

Road Race—a good 2-player road-race game with three separate racetracks (requires dual joysticks).

Seawolf—a shooting-gallery-type game; you try to sink various ships.

Tag—a 2-player game (requires dual joysticks).

Sketchpad and Maze—sketchpad allows drawing graphics with the joystick; Maze tests your maneuverability through a maze.

All programs run on any model PET. Several are written in machine language or have machine-language routines for faster graphics motion and control.

TIS Workbooks

TIS Workbooks are valuable for newcomers to the PET. There are currently six volumes in the series covering the PET in detail with examples, sample programs and short exercises. Spaces are provided within the text to record notes and routines.

Since the books were written for the original 8K PET, errata sheets are now included for the new ROMs and the 16K/32K machines. The individual volumes sell for $3.95 to $4.95 each; the entire set is $19.95, plus shipping, from Total Information Services, PO Box 921, Los Alamos NM 87544.

Workbook 1—Getting Started with Your PET, written primarily for people with little computer programming experience. It tries to provide enough information to allow the reader to input, run, save and load programs. It introduces the fundamentals of PET BASIC: calculator and program modes, data input and output, data representation and program storage on cassette.

One obvious error appears on pages 6-4 and 6-5. Several examples determine the amount of memory space used by different BASIC variables. Since arrays are used, the manual incorrectly states that floating-point variables take five bytes, integers take two and strings take three. This is only true for elements within an array. Individual variables are always seven bytes regardless of the variable type. Using a single integer variable (B%) uses the same space as a single floating-point variable (B). In fact, you can actually waste space by using the extra percent signs whenever the variable name is used in the program.

Workbook 2—PET String and Array Handling describes the features and limitations of arrays and subscripted variables. It covers string operations and related string functions in detail. Problem exercises help you better understand strings and arrays, while the index gives a detailed description of the various PET "character sets."

One word of caution for those with an older 8K PET and the original ROM set: pages 4-5 and 4-6 of this volume do not warn of the illegal use of a zero length in the LEFT$ and RIGHT$ string functions. This will result in an error on the old ROM set. The newer PETs and the upgrade ROM for the 8K PET now allow zero lengths in these functions without causing an error.

Workbook 3—PET Graphics covers the cursor-control characters, character sets, low- and high-density plotting, bar graphs, sketching and reverse video. The index includes complete listings for several programs developed in pieces throughout the workbook.

Workbook 4—PET Cassette reviews loading and saving program files and discusses all aspects of data file handling. It covers the OPEN and CLOSE commands along with transferring string and numeric data to and from tape data files. Status checking is also included with two application programs and information on cassette performance.

(continued on page 18)

MicroQuote

Your personal computer becomes a window on Wall Street.

MicroNET, the personal computer service of CompuServe, now offers MicroQuote, a comprehensive securities information system.

With MicroQuote you can gain information from a data bank of over 32,000 stocks, bonds and options from the New York, American, OTC and major regional markets plus Chicago options. MicroQuote contains price and volume data from January, 1974 with cumulative adjustment factors and dividend information from January, 1968.

You can determine indicated annual dividends, earnings per share, shares outstanding, BETA factors, open interest on options and amount outstanding on debt issues. MicroQuote can provide issue histories on a daily, weekly or monthly basis and even performs certain statistical analyses on the data. It's a vital tool for any investor.

It's just part of the MicroNET service

MicroNET also allows error-free downloading of software via the new software exchange and executive programs (now available for the TRS-80,® Apple II® and CP/M® systems). It also provides electronic mail service and can be accessed with a 300 baud modem via local phone calls in more than 175 U.S. cities. Write for full details on how your microcomputer can control one of the nation's largest and most sophisticated time-sharing computer centers for about 8 cents a minute!

TRS-80 is a registered trademark of Tandy Corporation
Apple II is a registered trademark of Apple Computer, Inc.
CP/M is a registered trademark of Digital Research

Regional distributors and local dealers wanted. Inquire to Dept. R

Software authors: MicroNET seeks to license quality programs for software exchange. Write to Dept. S

MicroNET™

Mail to: **CompuServe** Dept: K ✓147
Personal Computing Division
5000 Arlington Centre Blvd.
Columbus, Ohio 43220

Edited by Dennis Brisson

NEW PRODUCTS

Lowercase for Apple II

Reviewed by Art Little, ISI staff

Now you can add uppercase/lowercase capability to your Apple II with the newly released Keyboard Expandor. Unlike other products that use such keys as ESCAPE or CONTROL to indicate case, this hardware/software modification actually allows you to use the SHIFT keys, just as on a conventional typewriter.

The hardware change to the Apple II is a simple one-wire modification requiring a single solder point. The software modification is a totally transparent machine-language routine that augments the capabilities of the monitor while avoiding its uppercase conversion code. This routine permits the use of all Apple II characters and editing functions of the monitor as well as a capitals lock and shift lock. The software occupies only 1/4K and can be used on any size system; there are no system constraints with this product.

It is compatible with general display techniques, such as Paymar's LCA or Apple's contributed high resolution character generator, that display ASCII correctly. An Inverse mode option is included for those who do not already have a display method for uppercase/lowercase. The Keyboard Expandor is also totally compatible with DOS, allowing the use of uppercase/lowercase in TEXT files, PRINT and REM statements within BASIC programs and DOS file names, as well as in Immediate mode. Full documentation of both the hardware and software modification, as well as a floppy disk containing copies of the routines, is included with the package. Price is $20.

C & H Micro, PO Box 249, Clifton Park NY 12056. Reader Service number 477.

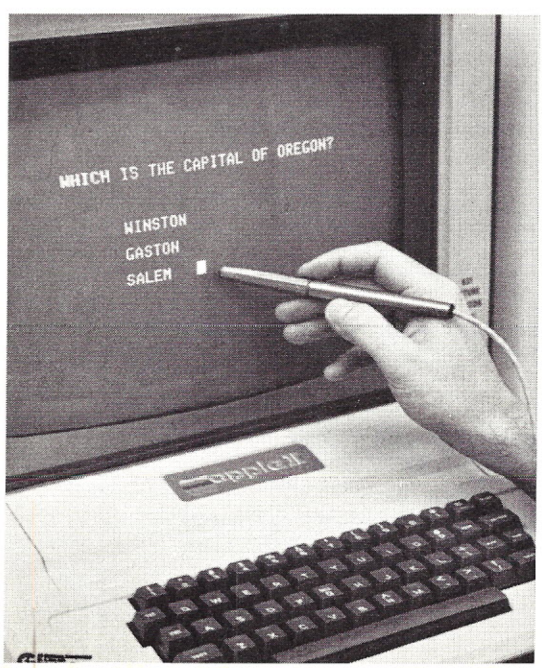
Light Pen for Apple II

A self-contained light pen that plugs directly into the Apple has been released by the 3-G Company, Rt. 3, Box 28a, Gaston OR 97119. The 3-G Light Pen makes it possible to bypass the Apple's keyboard and interact directly with the information displayed on the CRT screen. The light pen adds versatility to most graphics programs and makes possible unique games.

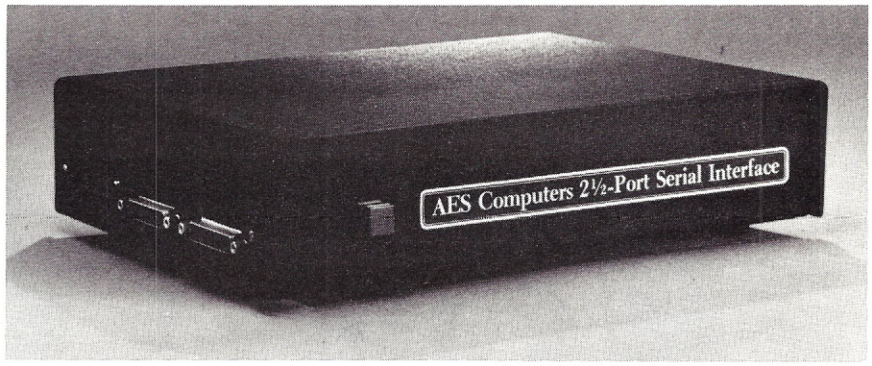
The user can make a selection from a menu displayed on the screen by using the light pen. This type of interaction makes it easy for the non-computer-oriented person to use an applications program. Another use of the pen is in educational programs as a teaching aid or a game for a young child.

The 3-G Light Pen is completely assembled and ready to plug into the Apple game paddle port. A demonstration game cassette, sample program and complete programming instructions are included with the pen. Price is $32.95, plus $1.50 for postage and handling; $6 for foreign orders. Reader Service number 480.

Apple II light pen at work.



Serial Interfaces



AES Computer's 2½ Port Serial Interface.

AES Computers, 118 S. Loara, Anaheim CA 92802, has introduced 2½ and 4½ Port Serial Interfaces, which provide two or four serial communication controllers, respectively. A flexible interface bus to the host computer can be adapted to various microprocessors with plug-in modules and adapter cables for TRS-80, Apple II, Exorciser, S-100, Multibus, STD Bus and others. The self-contained unit will operate 8080, Z-80, 6800 and 6502 in memory-mapped or I/O-mapped environments. Using the 8251A serial controllers, the interface supports synchronous communication modes to 64K baud and asynchronous communication modes to 19.2K baud. All aspects of the communication mode for each channel are software programmable.

The serial interface permits rapid implementation of advanced serial communication techniques for small computers. By providing two RS-232C interfaces and one isolated current-loop interface, the 2½ Port ($395) unit quickly expands small computers to build multiterminal or multiprocessor systems suitable for business, scientific and engineering applications. The 4½ Port ($695) supports four RS-232C interfaces and two current-loop interfaces. Reader Service number 478.

Studio II Conversion Package

The Studio II conversion package is designed to allow the owner of an RCA Studio II video game to convert his game unit into a simple microcomputer. The package consists of a PROM card, a RAM card, a backplane card, all the instructions necessary to install and operate the unit and six issues of the Studio II user's newsletter, *MicroStudio News*.

The backplane plugs into the Studio II game cartridge slot, and the PROM and RAM cards plug into the backplane. The four connectors on the backplane are mounted but not soldered. The PROM card and RAM card are completely assembled (except for

two 2114 RAM ICs) and tested. Two signals must be brought out from the Studio II for the RAM card. The package provides 1560 bytes of RAM and 512 bytes of ROM and includes the pre-programmed PROM containing the Monitor program. Price is $160.

ARESCO, PO Box 1142, Columbia MD 21044. Reader Service number 489.

Disk Protection

The C S S N BACKUP subsystem is a complete hardware/software package for the protection of disk-stored data. The off-line storage medium is a 13.4 megabytes capacity magnetic tape cartridge, making BACKUP ideal for use with high-capacity Winchester disks. BACKUP also offers the advantage of file-by-file backup, which allows you to restore just those files which were lost, rather than the entire disk.

BACKUP comes in a rack-mounted unit with a Z-80A/S-100-compatible interface board, a DEI cartridge tape drive and a CP/M-compatible software utility, featuring the file-by-file SAVE and RESTORE commands. Price is $2995.

C S S N, 120 Boylston St., 4th Floor, Boston MA 02116. Reader Service number 493.

TRS-80 Speech-Recognition System

Now TRS-80 users can program their computers to respond to spoken words with the TRS-80 Voxbox, a speech-recognition system from Radio Shack, 1300 One Tandy Center, Fort Worth TX 76102. Words or phrases may be used to enter data, control and instruct the TRS-80 without having to type on the keyboard. The Voxbox can be programmed with up to 32 words. The user decides what words are to be used, and what they do (the action caused or data provided) is written into the program.

The Voxbox should correctly recognize 85–95 percent of the words used, provided the user speaks clearly and distinctly. However, Radio Shack recommends that the unit be used primarily for entertainment and experimentation. The unit requires Level II 16K RAM and cassette recorder. A machine-language "driver" program and three demonstration programs, owner's manual and a push-to-talk dynamic microphone are supplied with the unit. Price is $169.95. Reader Service number 475.

AIM-65 Card File

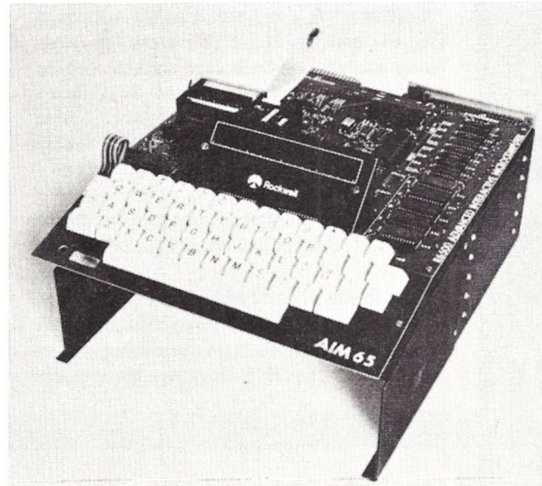
The MTU K-1005A-A Card File integrates the AIM-65 computer, keyboard and a series of expansion boards into one compact, portable unit, complete with motherboard. Drawing no power, the unbuffered motherboard utilizes the AIM bus structure to carry expansion connector signals to up to four additional boards. A fifth undedicated position is provided for a board not on the bus.

The card file features a U-shaped black anodized aluminum frame measuring 15½ × 11½ × 4½ inches (exclusive of AIM). To keep the expansion bus noise-free, the motherboard is a double-sided printed circuit board with groundplane. Price is $95, including motherboard and user manual. An applications motherboard ($29) is optional. Other card files are offered for the PET, KIM-1 and SYM-1 computers.

Micro Technology Unlimited, PO Box 4596, Manchester NH 03108. Reader Service number 476.

Apple Carrying Case

A rugged, custom-designed Apple computer system carrying case is now available from Computer Textile, Inc., 10960 Wilshire Blvd., Suite 1504, Los Angeles CA 90024.



Micro Tech's card file for the AIM-65.

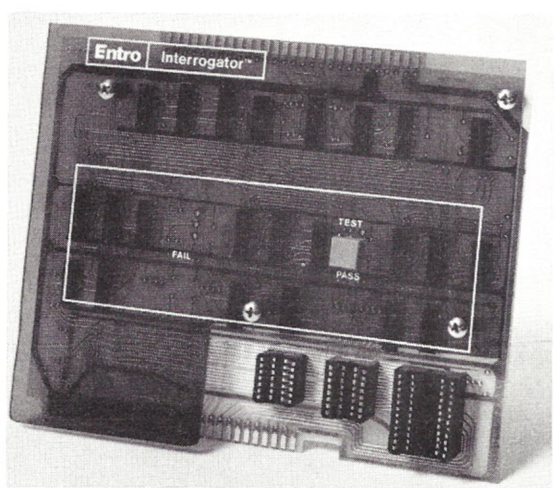
The case contains room for an Apple, 9 inch Sanyo monitor, two disk drives, power strip, two boxes of diskettes and manuals.

The case is finished in black vinyl with metal-reinforced corners. The interior is lined with protective foam rubber covered with black velveteen for a professional appearance. The carrying case includes a built-in shelf over the Apple for the monitor and disk drives so the system may be operated in the case, which measures 30 × 21½ × 10¾ inches and weighs approximately 12 pounds. Price is $199. Reader Service number 487.

IC Tester

The 7310 Interrogator tests all standard and custom 5 V digital integrated circuits, 7-segment LEDs and even small circuit boards with up to 24 leads or test points. The unit is driven from a single eight-line output port of a host computer and can exercise and test either individual ICs or groups of ICs in a system under test.

Radio Shack's TRS-80 Voxbox.



Entro 7310 Interrogator IC Tester.

Individual ICs are tested in the on-board 14-, 16- and 24-leaded ZIF (zero-insertion-force) sockets. Nonstandard sockets and/or worst-case output loads connect through the 26-lead output expansion connector. Active loads on all test pins assure thorough testing of devices with Tri-state or open collector outputs. A separate power bus allows optional manipulation and monitoring of the device under test for special or worst-case conditions.

Test complete and fail test LEDs, as well as start test button on the Interrogator, are ideal for use in production line or incoming inspection modes. The 7310 Interrogator operates with any computer containing at least 1024 bytes of RAM memory and eight TTL output lines. Price is $249, including testing manual.

Entro, 1171 Borregas, Sunnyvale CA 94086. Reader Service number 481.

Switching and Monitoring Units

Giltronix, Inc., 450 San Antonio Ave., Suite 44, Palo Alto CA 94306, introduces a new family of switching and monitoring units for interfacing, configuring and monitoring computers, terminals, printers and modems, as well as other devices complying with the RS-232 and the IEEE 488 specifications:

- The GRS 232-P24 switches 24 pins. Pin 25 of the 25-pin EIA connector is not switched.
- The GRS 232-S24 switches 24 pins. Pin 1 of the 25-pin EIA connector is not switched but is common to all four EIA connectors.
- The GRS 232-2P24 is similar to the P24 but contains two units in the same housing. It will allow switching of five devices.
- The GRS 232-2S24 is similar to the S24 but contains two units in the same housing.

Each unit consists of a standard three-way switching system and an optional interface monitor. Reader Service number 486.

Edited by Dennis Brisson

NEW SOFTWARE

Program for Businessmen

How can a computer serve your business? Heath Company, Benton Harbor MI 49022, can help you answer this question with its latest program, "Computer Concepts for Small Business," designed to give the small businessman the information and understanding needed to evaluate how a computer can benefit his business.

The program includes three audio cassettes, which guide the reader through the text material, highlighting important points and making this program more than just a textbook reading experience. The 160-page illustrated workbook describes the types of memory in a computer, compares the capabilities of different types of storage media and I/Os and discusses the types of software and the tasks they can be designed to perform in a typical small business.

The program also covers the types of personnel needed to run a computer and describes the advantages of time-sharing vs service bureau vs owning. It compares the types of computers and how to select the one that best serves your needs. It even tells you how to select the right dealer to buy your computer from. Price is $49.95. Reader Service number 479.

Heath's program to evaluate computers in business.

EPROM Programmer Software for PET

Optimal Technology, Inc., Blue Wood 127, Earlysville VA 22936, announces EPROM Programmer software for the PET using OT's Model EP-2A-79 EPROM Programmer. Software, supplied on cassette, includes routines for checking if the EPROM is erased, programming, verifying programming and reading the contents of the EPROM into memory. The programmer connects to the PET via the user I/O port. The program is self-prompting. Software supports the programming of a variety of EPROMs, including the 2708, 2716, TMS 2716, 2732 and the new Motorola 8K×8 MCM68764 EPROM. Price is $19.95. Reader Service number 485.

CP/M BASEX Compiler

BASEX interactive compiler and loader programs include extensions to allow full use of all CP/M operating system facilities and include commands to save and load complete files and list disk directories. BASEX, a new intermediate-level language for microcomputers that combines some of the best features of both BASic and EXecutable machine-language code, allows you to enter, list, edit and run your program without the help of any auxiliary programs such as editors or linkage editors. Programs run up to 20 times faster than similar BASIC programs. Also,

the BASEX run-time routines are only 2K bytes long, and thus typically require approximately 6K bytes less memory than similar programs that are run with an 8K BASIC interpreter.

Most BASEX commands and features resemble their counterparts in BASIC. In addition, BASEX allows variable names of any length, block memory searches, block memory transfers and named subroutines that can pass multiple arguments to and from the calling program. The $43 price includes manual, CP/M addendum and 8 inch CP/M disk.

Interactive Microware, Inc., PO Box 771, State College PA 16801. Reader Service number 497.

Apple II Time Calculations

Almanac, a new Apple II program with functions related to time, the calendar and general astronomy, should interest those involved in time calculations—general users, businessmen, amateur astronomers, shortwave radio listeners and time travelers. Functions include:

- Calendar calculations that include printing calendar pages for any month specified—even prior to the Gregorian calendar reform—calculation of day of week and day of year for any date and calculation of date of Easter.
- Sidereal time, including a software sidereal clock for calculating star time.
- Sunrise and sunset calculations.
- Calculation of phases of the moon and the dates and times of solar and lunar eclipses.
- Solar system model showing the relative positions of the planets for any date and illustrating the rotation of the planets around the sun, while displaying the time in hundredths of a year.
- A real-time clock that shows the time in time zones around the world.

The program requires at least 32K RAM, Disk II and Applesoft II in ROM. Price is $29.95 for disk.

Williamsville Publishing Company, PO Box 250, Fredonia NY 14063. Reader Service number 496.

Apple II Information System

Infotree is an interactive information storage and retrieval program written in UCSD Pascal for the Apple II. It allows you to store and recall information and organize it in any manner desired. All data stored by Infotree is referenced by a key, a string of up to 18 characters. Up to 18 lines of text can be associated with any key. In addition, any key can own a list of sub-keys, allowing you to separate data into different logical groups to any level desired, as any sub-key can own its own list of sub-keys. Infotree is capable of storing 4380 records on a single diskette.

To run the system press the first letter of the command desired. Infotree fills in the rest and then asks for the key(s) it needs to execute the command. Pressing → recalls the keys used in the last command one at a time. Infotree also automatically formats the keys as they are typed in. Typical applications include the storage of personal information, mailing lists, phone number directories, appointments scheduling and memos. Price, including user's manual and tutorial, is $39.95.

Siro-Tech, 6 Main Street, Ogdensburg NY 13669. Reader Service number 482.

Statistical Package

Microstat is an advanced statistical package that uses special algorithms designed to minimize errors introduced into many statistical calculations using large numbers. Designed for serious scientific, research and business applications, it utilizes a data management subsystem (DMS) to control, edit and modify all files that are used as data input into the system.

Designed for use on 8-bit computers, Microstat features eight probability distributions, eleven non-parametric tests, chi-square, one- and two-way ANOVA, hypothesis tests (mean and proportions), simple and multiple regression, time series (including exponential smoothing) and data plots. The package is for use with the North Star Disk Operating System and BASIC, one or two drives. Price is $200; if purchased separately, manual costs $10.

Ecosoft, PO Box 68602, Indianapolis IN 46260. Reader Service number 484.

Data Management System

ICDMS is a new data management system from International Cybernetics, 8 Mary Dr., RR 7, Bloomington IL 61701. This menu-driven program allows you to define up to ten numeric or alphanumeric fields for file records. Record functions include: ADD, CHANGE, DELETE, SORT, SAVE, LOAD and LIST, with many of the functions operating on one, all or a range of records in the file. Field functions allow you to SUM numeric fields or SEARCH for any two to 256 character combinations. In addition, INTELLIGENT SUM combines the search and sum functions to sum numeric fields in records meeting search qualifications.

The program is available for the Heath H-89 and Radio Shack TRS-80. Price is $49. Reader Service number 495.

Real Estate Program

Property Analysis System (PAS), for both residential and income property, analyzes the effects of financing, income, expenses, depreciation, taxes and inflation on the return on investment for nine years. PAS produces a three-page analysis consisting of initial conditions and a nine-year projection of property value, liabilities, equity, gross income, expenses, net spendable income and the cash and percentage return on investment before and after taxes. The five-program system written in Micropolis BASIC requires 48K bytes of memory and a CRT with cursor control. It comes initialized for a Hazeltine 1500 CRT and a Centronics 779 printer. Price is $195.

Investment Analysis Systems, PO Box 282, Palos Verdes Est. CA 90274. Reader Service number 492.

More Software

Series 8000 Medical and Dental Management Systems Upgrade—from Univair, Inc., 10327 Lambert Intl. Airport, St. Louis MO 63145. New features include optional default of current date for all tickets and reports, carriage-return defaults for all editing functions, patient's name displayed when entering tickets and payments and combined ticket and loading routines. Designed for the TRS-80 Model II and most 32K CP/M disk-based microcomputers. Price is $495. Reader Service number 488.

Disk Fix—a general-purpose disk utility for Mits/Pertec disks that allows any sector of an unmounted diskette to be examined, edited and/or rewritten. Requires 8080 mainframe with 32K, a Mits/Pertec floppy disk drive, CRT and Mits/Pertec disk extended BASIC. Price is $95. Software Store Ltd., 706 Chippewa Square, Marquette MI 49855. Reader Service number 498.

Temple of Apshai—a solo fantasy adventure game for the Apple that lets the player lead an alter ego on an adventure into a labyrinth of over 200 caverns and chambers, all graphically illustrated in color. The degree of difficulty may be adjusted. Price is $24.95 for cassette and $29.95 for disk. Both versions require 48K and Applesoft. Automated Simulations, PO Box 4232, Mountain View CA 94040. Reader Service number 490.

TRS-80 Business Programs—Pro Forma (forecasted) Cash-Flow Budget program plans a company's cash needs for up to 12 periods. Price is $125. Lease vs Purchase program evaluates the benefits of leasing as opposed to purchasing an asset, in light of the newest tax laws. Price is $100. Management Systems Software, Inc., 5200 Brittany Drive, #1006, St. Petersburg FL 33715. Reader Service number 491.

Extended Memory Multi-User Software—EFAMOS is a new multitasking DOS for 8080, 8085 and Z-80 microcomputers that supports multitasking and multi-users with memory mapping. Up to 3 megabytes of memory can be available to users through 32K memory banks. EFAMOS with BASIC compiler, assemblers, utilities and word-processing software is available to OEMs and dealers. MVT Microcomputer System, Inc., 9241 Reseda Boulevard, Suite 203, Northridge CA 91324. Reader Service number 494.

BOOK REVIEWS

6502 Assembly Language Programming

Lance A. Leventhal
Osborne/McGraw-Hill
Berkeley CA, 1979
Softcover, $9.95

A text on assembly-language programming for microprocessors may address several distinct audiences. Commercial users require training manuals for engineers, both those new to microcomputer design and those seeking an introduction to a new processor. Hobbyists learn assembly language out of curiosity, and to squeeze additional performance from their computers. Personal and business computer owners, using appliance computers, and the growing number of scientists and engineers incorporating personal-type computers into larger systems may be forced to use assembly language to implement functions not provided by the system's designers. The interests and technical background of these groups differ greatly, and are gradually being reflected in the growing supply of textbooks and manuals.

Leventhal's *6502 Assembly Language Programming*, following the format and comprehensive approach of the *Introduction to Microcomputers* series also published by Osborne/McGraw-Hill, succeeds in addressing all these audiences within the space of a single volume. Leventhal adopts a clear, unadorned style that instructs the reader in understanding microprocessor jargon without relying on that jargon. The book avoids excessive abbreviations and acronyms, but the reader will find definitions and examples of their use sufficient to make him comfortable in reading both hobby journals and trade magazines.

This is a large book, almost 600 pages. Extensive sections are printed in boldface letters, clearly and usefully distinguishing basic information from discussion of technical details. The novice may omit lightface sections without losing continuity, while the advanced reader may use these sections as a key, skipping material with which he is already familiar. Abundant figures, tables and illustrations amplify and summarize difficult points; Leventhal uses figures for instruction, not decoration, and includes few superfluous figures even though an illustration adorns almost every page of the text.

The coverage of the text is exceptionally complete. The first two chapters discuss the definition of assembly language, problems for which it may be suitable and the syntax of common assemblers. The standard MOS Technology assembler syntax is used throughout the book. Chapter 3 discusses each operation in detail and includes valuable overviews of the entire instruction set and of the various addressing modes available to the programmer. The use of indirect addressing, often a confusing topic, is succinctly discussed. Chapter 3 alone is over 100 pages, but prudent pagination and bold page headings simplify rapid location of individual discussions for later reference.

Succeeding chapters treat specific programming techniques and program structures. Loops, string handling, arithmetic routines and the use of tables and list structures are among the topics receiving special consideration. Each chapter includes several specific examples, tested by the author, and concludes with problems extending the worked examples. These problems are suitable for classroom instruction or for independent work and might make the book especially attractive to teachers. Flowcharts, listings and sample data are provided for each example, and the additional references cited in these chapters form a convenient introduction to the literature through mid-1979.

Chapter 11 extends the discussion to input and output devices. Again, the discussion exceeds 100 pages and includes numerous examples. Simple but useful devices, including LEDs, 7-segment displays and keyboards, form a foundation, and Leventhal considers their use at great length. The important LSI support devices commonly used with 6502 microprocessors, including parallel, serial and multiple function devices, are discussed at length. The emphasis throughout is on software aspects of system design; hardware discussion is kept to a minimum.

The concluding chapters treat the important questions of program design and organization with sophistication. Leventhal includes an excellent discussion of the advantages and the costs of modular and structured program design, and of top-down development procedures. His treatment of documentations sets a standard which (unfortunately) few manufacturers match, but which should permit programs to be used and effectively maintained for years to come. In a concluding chapter Leventhal illustrates this design regimen by discussing two moderately complex projects, implementing a digital stopwatch and a thermometer.

Despite its dimensions, *6502 Assembly Language Programming* suffers from some omissions. There is almost no discussion of the applicability of the 6502 to various tasks, or comparison of this processor with other available processors. The 6502 appears to be especially well suited for certain tasks, as witnessed by the greater speed of most 6502 BASIC interpreters compared with, say, their 8080 rivals; when may the designer expect to take advantage of this speed? Similarly, Leventhal provides little guidance to the designer who doubts the feasibility of using a 6502 in his design; although the discussions of program redesign to conserve time and memory are excellent, little is done to prevent the costly attempt to implement impossibly high-speed or complex algorithms.

Finally, Leventhal's attempt to rigorously separate hardware from software strikes this reviewer as inherently artificial, even though excellent discussions of hardware considerations are available elsewhere. Placing small or moderate amounts of logic in hardware or software often has a great impact on the design of software systems; programmers should be encouraged to consider implementing functions in hardware as well as to take logical functions over into the program. Although this is not very relevant to the personal computer/business computer audience, the commercial designer should know when to ask his engineering staff to modify the hardware design, if possible, in order to make his own task simpler and more reliable.

6502 Assembly Language Programming is relatively free from typographical errors. It does not expect extensive technical background on the part of its readers, although the novice might be advised to follow the author's advice and obtain a copy of the first volume of Adam Osborne's *An Introduction to Microprocessors* (Osborne/McGraw-Hill, Berkeley CA). Few programmers and system designers will fail to find new and useful information in this book; many will find it a useful permanent reference.

Mark Bernstein
Cambridge MA

8080/8085 Software Design—Book 2

Titus, Larsen, Titus
Howard W. Sams & Co., Inc.
Indianapolis IN
Softcover, 348 pp., $9.95

Having read and learned a great deal from *8080/8085 Software Design* (see *Microcomputing*, September 1979, p. 12), I was pleased to see that it was only an introduction to software design for the 8080 and the 8085. In the preface to *8080/8085 Software Design—Book 2*, the authors state that, while not required, the first volume is an excellent reference for the second volume. I urge purchasing the books as a set. The second volume makes numerous references to the first.

With the multitude of high-level languages available to the microcomputer user, why should you buy a book written at assembly-language level? The main theme of the two books, especially the second, is software design. Chapters 4 through 10 qualify the book as a computer-science source book. Should you de-

Now! North Star Application Software!

North Star now offers application software for use on the HORIZON! Now you have one reliable source for both hardware and software needs! The first packages available are:

NorthWord—
NorthWord is a simple-to-operate word processing system designed for use with the popular North Star HORIZON. NorthWord enables you to increase office efficiency and cut document typing time and cost. NorthWord incorporates the most sought-after word processing features: easy editing, on-screen text formatting, simultaneous document printing, and much more. NorthWord can be integrated with other North Star software packages to produce customized letters, labels and reports quickly and efficiently.

MailManager—
MailManager enables you to compile and maintain complete organized mailing lists. Lists are easily accessible and can be compiled with a great deal of flexibility. Entries, corrections and deletions are easily made. The North Star MailManager can print your list on individual envelopes, on mailing labels, or in compact summary form.

InfoManager—
InfoManager is a powerful list-oriented, data management system. It will accept up to 50 categories of information for each record and has the ability to select and sort before printing. The North Star InfoManager has power and flexibility for many applications: product inquiry, inventory, customer/client records, calendar reminders, and as an easy way to fill in often-used forms.

GeneralLedger—
General Ledger and Financial Reporting, two programs in one, maintains general ledger accounts based on such input as checks, bank deposits and journal entries, and uses the information in the general ledger to produce customized financial statements and financial reports.

NorthWord is the central building block for all the North Star application software to follow. Packages now being tested include other accounting and professional application packages. For more information or a demonstration, contact your local North Star dealer.

NorthStar
North Star Computers, Inc.
1440 Fourth Street
Berkeley, CA 94710
(415) 527-6950
TWX/Telex 910-366-7001

sire to write your own I/O drivers, interrupt service routines, system monitor or debuggers, you can do so with the information from chapters 1 through 3.

Chapter 1 deals with serial communications, both hardware- and software-based. A discussion of serial data formats precedes the hardware discussion, which centers around two serial communication devices: the universal asynchronous receiver/transmitter (UART) and the universal synchronous/asynchronous receiver/transmitter (USART).

Chapter 2 covers interrupts—what they are, the different types, how to use them, how to prioritize them and why to avoid them. For the intrepid user of interrupts, chapter 3 gives both software and hardware applications.

Chapter 4 has a concise discussion of data structures, which any programmer who wants to do data manipulation should read. This chapter is the first of four that deal with data structures and how they are accessed.

Chapter 5 expands upon the theory of chapter 4 by illustrating search subroutines with numerous examples. Chapter 6 deals with sorting numeric values and alphanumeric strings. Two methods are used: the straight insertion sort and the exchange (bubble) sort. Look-up tables are the subject of chapter 7.

Chapter 8 by itself is worth the purchase price of the book. It deals with command decoders for both fixed-length and variable-length commands. Copy the variable-length command decoder for your system monitor, and you won't have to worry about writing another command-decoder program.

System monitors are covered in chapter 9, and the last chapter covers breakpoints and debuggers.

Two appendices are included. Appendix A is a reprint of the Mostek MK5009P data sheet, and appendix B contains some final thoughts on microcomputer interfacing.

Some final thoughts on the book. In the preface, the authors stated that a goal of the book was to provide detailed descriptions of how the programs work. In my opinion, they succeeded very well—the book could be considered either a computer-science book for programmers interested in the theory of data structures or an assembly-language programming book for 8080/8085 programmers who want to copy programs from the book and have them work. Every program I have used from either book has worked perfectly (hand-assembly errors aside). I recommend that all 8080/8085 programmers buy the book. I make the same recommendation to programmers using other microprocessors who desire a better understanding of data manipulation. You will be able to understand how the programs operate even without a detailed knowledge of the 8080 instruction set.

J. C. Hassall
Blacksburg VA

On Line

D. H. Beetle, Publisher
24695 Santa Cruz Hwy., Los Gatos CA

Touted as the "only nationwide classified advertising newsletter devoted entirely to the computer hobby," *On Line* is an unusual publication. Published 17.3925 times per year (every third Friday), *On Line* is a 40-page "buy and sell forum for the computer hobbyist" that is into its fourth year of life. Printed in a 6 × 9 inch pulp format, *On Line* contains individual and commercial advertisements, mostly full width, right-justified and easy to read. There is also some display-ad material from major national vendors (Lifeboat Associates, Percom, Electrolabs, Electronic Systems).

One of the features that makes *On Line* unusual is its advertisements. You'll find unexpected items: an index to computer magazine articles, speedy but reasonably priced PROM programming, lots of special-interest newsletters, programs seldom, if ever, seen before, microcomputing books, how to "take full advantage of your phone line," hobbyist sources for continuous forms and other supplies, swap-meet announcements and a computer-assisted Bible-study course.

Other distinguishing *On Line* features are in policy and format. At the policy level, the publisher distinguishes between commercial and hobbyist advertisements: a hobbyist ad for one issue costs $1.50 per 80-character line, and a commercial ad costs $3.50 per line. These rates are for subscribers (nonsubscribers pay a buck more per line) and go down if your ad runs in more than one issue.

On Line provides free advertising space to "people offering useful information/fixes/programs/reference lists or cards available free or for $1 or less" and to "people offering to serve as a focus for local club or nonbusiness-related users group ads announcing meetings/nonprofit events."

A full page is devoted to computer club meeting schedules—probably the most readable (but not complete) listing available. About 100 clubs are listed. Format features include each page vertically indexed by ad number and horizontally indexed by column number (1–80) plus page number, date, volume and issue numbers. Each issue gives a schedule of upcoming issues with "ads due by" and when they will appear.

A few shortcomings exist. The advertisements are not classified (the publisher does indicate this will be rectified). A few advertisements are so brief that they're hard to understand. For the most part though, *On Line* is a publication that seeks to promote the hobby computer world. It is even priced for hobbyists: four issues cost $1, 18 issues are $3.75 and 36 issues are $7, all mailed first class.

Douglas Haden
Socorro NM

COMPUTER CLINIC

I am an Englishman living in Argentina. I own an Apple II System (48K, two floppy-disk units). If anyone would like to correspond, I would be pleased (sort of an "Apple out on a limb!").

P. V. Korrison
Murature 228
Punta Alta 8109
Republica Argentina

A project of mine requires a special display, one that probably does *not* exist.

Can anyone recommend some companies that might be able to make one for me? I need 360 evenly spaced, independently selectable LEDs arranged in a circle not larger than three inches in diameter, and since only one light will be on at any time, TTL logic will be used to drive the thing. My breadboarding has reached, what seems to be, a permanent "wait state."

Joseph A. Lindo
Box 1714
APO New York 09109

We had a small house fire. Destroyed were the schematics for a Wave Mate III Z-80 microcomputer system. In writing Wave Mate, I found they had gone out of business. Does anyone know where I might purchase a set of schematics? Where do you get software for a system that is no longer manufactured?

G. Johnson
Winhall Craft Distributor
PO Box 9-A
Bondville VT 05340

I am restoring a Raytheon 703 minicomputer to use as my personal computer for engineering design and simulation studies. I am looking for readers who are current users or knowledgeable about the machine and will share information.

Rudolf F. Wrobel
12725 West 55th Terrace
Shawnee KS 66216

LETTERS TO THE EDITOR

Praise

It is rare for me to write to a magazine with either praise or complaint, but I am long overdue. I have the complete run of *Kilobaud/Microcomputing*, and I receive *Byte*, *Interface Age* and *68 Micro*.

I must praise the articles by Peter Stark. They are the best things I have seen in any of the magazines to date. I am one of those "computer experimenters" who are becoming less and less heard from. My system is an SWTP 6800/2 with 20K, Percom disk, CT-64 and printer. Peter's articles have been the biggest help I have received, starting with Kilobaud Klassroom and now his SWTP system notes . . . great! It is rare to find an author who is easy to read and has such a good grasp of his material. He has saved me from some dead ends and costly mistakes. I have recently renewed my subscription. One reason is Peter's articles; the other reason is your treatment of the 6800. You publish the best journal for us small-systems people.

Dennis Doonan
Racine WI

CP/M and You, and You, and You . . .

Because of the volume of mail, both pro and con, author Thom Hogan received on "CP/M and You" (February 1980, p. 183), he has written this all-inclusive reply.—Editors.

Many computer users felt strongly enough about my article, "CP/M and You," to write me or *Microcomputing* about it. Half of you wrote to express your thanks for the article, while the other half wrote strongly degrading both my writing and my opinions about CP/M.

I have yet to meet the person who has picked up the CP/M manuals and figured out the system in one reading. I was not even close. For those of you who question my motivation and my knowledge: I work full time managing one of the Midwest's most successful computer stores. My motivation for writing the article was to put to rest the notion that computer users will tolerate sloppy or unintelligible documentation or mental gymnastics necessary for use of software. CP/M is indeed the best operating system currently available for most computers. At the same time, I have had to teach well over 100 customers how to use it. Does that make any sense?

Why should you have everything provided by Digital Research on your diskette? Because most users can't get the programs off. I know that sounds facetious, but it is mostly true. I have actually seen people hand a "non-computer type" a CP/M diskette and a set of manuals and laugh. Seventy percent of the sales at our store are business systems. Ninety percent of those sales are for systems that use CP/M or a derivative. After fighting it, we finally gave up. Now we spend upwards of one man-week carefully creating all of the diskettes any business will need. The first thing we do is erase the unnecessary programs.

Program developers need to have all the utility programs available on-line. I don't know if you have noticed, but diskette wear increases with handling—and I generally have about six hours of programming a day to look forward to. I also will not tolerate a system that requires a ten-page listing just to find which diskette has the utility I'm looking for.

Using CP/M with one disk drive. I don't know why so many of you picked up on this point that I made in the article. I've come to the conclusion that one disk drive is not feasible under *any* operating system I have yet to see. I know you can use DDT to copy a program from one diskette to another using a single-drive CP/M system. This is not explained in the manuals, and I point that out on the second page of the article.

Why do I harp about I/O drivers when they come configured? They don't always come configured the way you want them. I wrote the article over a year ago, and a lot has changed since then. However, those of you who have bought Electric Pencil and a Diablo printer know that you can't run EP right away. You must change the I/O drivers to look for the Diablo's keyboard, even if it doesn't have one! That's why I made a point about I/O drivers.

My bottom line still stands. In fact, if anything, I now stand by it even more strongly:

- You must have two drives to use CP/M or any other DOS effectively.
- 48K is a necessity. For those of you who took me to task about this, take a look at the list of software you suggested as being available for use on CP/M: Structured Systems Group (requires 48K, two drives), Serendipity (requires 48K, two drives for large data files), Wordstar (strongly suggests 48K, two drives). The Pascals that are available generally take 56K. If you wish to use Digital Research's fine DESPOOL for simultaneous print, you'd better have some extra free space, also.
- You need at least ½ megabyte of disk storage.
- Expect to be confused. Digital Research wrote the manuals for system developers, not end-users. I'm in the process of writing a small booklet, which we will provide all users of our CP/M-based software and which we will provide with CP/M-equipped hardware.
- If the software you need isn't available under CP/M, don't expect it to be here tomorrow. Too many people are buying CP/M just to be "software compatible" and not because they need it. I currently have the capabilities of transferring *any* ASCII stream from one of the machines we sell to any other, and I don't need a software bus in that case.

If anyone has differing viewpoints on CP/M, I'd love to hear them. I like CP/M as an operating system. I dislike people buying things they don't need or won't figure out how to use. As a computer-store manager, I enjoy the challenge of trying to find the right equipment for the right customer.

Thom Hogan
Basically Speaking
719 Anna Lee Lane
Bloomington IN 47401

"NAVPROG" Right Up There

Congratulations to Leland Young for his article, "NAVPROG," in the February 1980 issue. In converting the program for use on my TRS-80, I have discovered one bug. Under certain conditions, the flight plan will give an incorrect magnetic heading (MH). We don't want any pilots getting lost! (See Example 1.)

As a commercial pilot with a major U.S. airline, I find that "NAVPROG" provides an insight into how our airline's computerized flight plans must be derived.

Paul Sturpe
First Officer, USAIR
Coraopolis PA

Elegant Elf EPROMmer

Many thanks for the article by Robert Cotter in the February 1980 issue describing an elegant 2708 programmer for the Elf II ("The Elf EPROMmer"). We built one on a 4 by 4½ inch card that plugs into the 44-pin connector on a

```
Change Line 1600 to read:  IF X>180 THEN X2=X:X=360-X
Change Line 1660 to read:  IF X2>180 THEN Q(I)=W4+X1:GOTO 1680
```

Example 1.

Super Elf board. The Super Elf 4K expansion board contains an output port that we attached to the EPROM board with a 16-pin DIP connector.

Instead of the 67 instructions required by Elf II, the Super Elf output port uses 63 instructions, and this substitution should be made in the program. We chose to input the N0 and N1 lines to an unused gate on the 7400, invert the output with 1/6th of the 7406 (a pull-up resistor is required) and substitute the resulting signal for N2. However, it should be sufficient to substitute N1 for the N2 shown in Fig. 1 of the article.

Fig. 1 contains a typographical error. The A9 input pin on the EPROM socket is incorrectly marked 24; it should be 22.

David Moews, Paul Moews
Storrs CT

DD*

Please consider yourselves to have been flogged with a wet towel. For shame! Your use of the phrase "Due to a computer error . . . " on page 18 of the February 1980 issue of a magazine widely read in the computer industry addresses your audience in a recognizable fashion . . . somewhat like waving a reg flag in front of a bull.

**Pair o' Ds. We were just poking a little ironic fun at the "wave the reg flag at the bull" approach. Concerning the towel-flogging, we're a little worried . . . we liked it.—Editors.*

Challenging the Thought Processes

The OSI Challenger II article by Richard Lary in the February 1980 issue of *Microcomputing* started me thinking about a problem I gave up on several times. The code given by Lary for quick screen clear has appeared in several publications, but the article stated that the routine was in ROM and could be located anywhere in memory without change. Nobody told me that before!

I finally located the code starting at 65036 and ending at 65061 (it had two extra bytes for some reason). I have never been able to add two numbers in machine language, but I was able to devise a way to transfer the existing routine to RAM, add the RTS and do the other necessary POKEs to make it go. The basic code is shown in Example 2. Line two PEEKs at one byte of ROM, calls it "y" and POKEs it back into RAM at a fixed distance (60967 bytes) away.

The advantage is a saving of memory space and typing because ROM instead of DATA is read. Programs that use a lot of scrolling to clear the screen can wind up with an overall saving in memory. I'm memory conscious because I haven't expanded from the original 4K . . . actually 3327 usable bytes.

OSI makes a fantastic machine, but documentation is nil. A book could be published about the goodies people have found by accident. Also, if anyone could figure out how to use the existing polled keyboard routine (with changes) to read the keyboard *without stopping* if no key were down, I could proceed with my long-desired CW sending program.

Robert D. Garrett
Decatur AL

```
30000 POKE 133,228; POKE 134,15; POKE 11,229;
      POKE 12,15; POKE 4095,96
30010 FOR M = 65036 TO 65061; Y = PEEK(M); POKE
      M - 60967,Y; NEXT
```

Example 2.

Necessary Evil

Allan Flippin's article in the March issue on assembly-language benchmarks was super. I appreciate the enormous effort that went into it. I must, however, contest, or at least qualify, some of his observations and conclusions.

For lack of a better method, benchmarking may be a necessary evil in rating processors. However, the real proof is performance in a real application where program length is on the order of 1K or more. In a real application, most of the processors tested will be hard-pressed to equal the memory utilization of either the 6502 or 1802, even though in the benchmarks they are rated only seven and eight. Since the architecture and instruction sets of both of these, particularly the 1802, depart significantly from the 8080 norm, even the most ardent aficionados often grossly underutilize their power.

Although Mr. Flippin allows the 9900 to have registers in memory space, he doesn't mention that with its indirect addressing capability the 6502 effectively has up to *128 pairs* of registers in memory that can be used as address pointers. The 8080 has the lone HL register, which becomes a serious bottleneck. Although 8080 memory reference instructions are only one byte long, the load-immediate takes three bytes, resulting in a total of four bytes. The 6502's memory references require two-byte instructions, but these register pointers can be table-loaded into zero page, with negligible load overhead, and left there indefinitely.

In short, the 6502 and 1802 do much better in memory utilization in real applications than the benchmarks indicate because both take advantage of clever programming techniques unique to their architecture which don't come into use in benchmark-level programs.

By knowing and understanding your processor's architecture and instruction set, and applying techniques to take advantage of its strong points, you can expect performance better than indicated by the benchmarks in real world applications.

Gene Zumchak
Buffalo NY

Please type all letters to the editor in uppercase and lowercase letters, and double-space the typing. Thank you.

PET-POURRI

(from page 8)

Since this volume was written for the older 8K PET, the errata sheet is extensive because of the original problems with data file handling. The newer models and those with the ROM upgrades no longer have to turn on the cassette motor to manually create wider gaps between tape records. The operating system has been corrected to automatically fix this annoying problem. However, this volume should be valuable in understanding the use and implementation of data files under program control.

Workbook 5—PET Miscellaneous covers unrelated features and functions not covered in the other volumes. The errata sheet for this volume does not give the new values for the material on pages 5-4 to 5-9. The primary floating-point accumulator is now located at locations 94 to 99 (decimal).

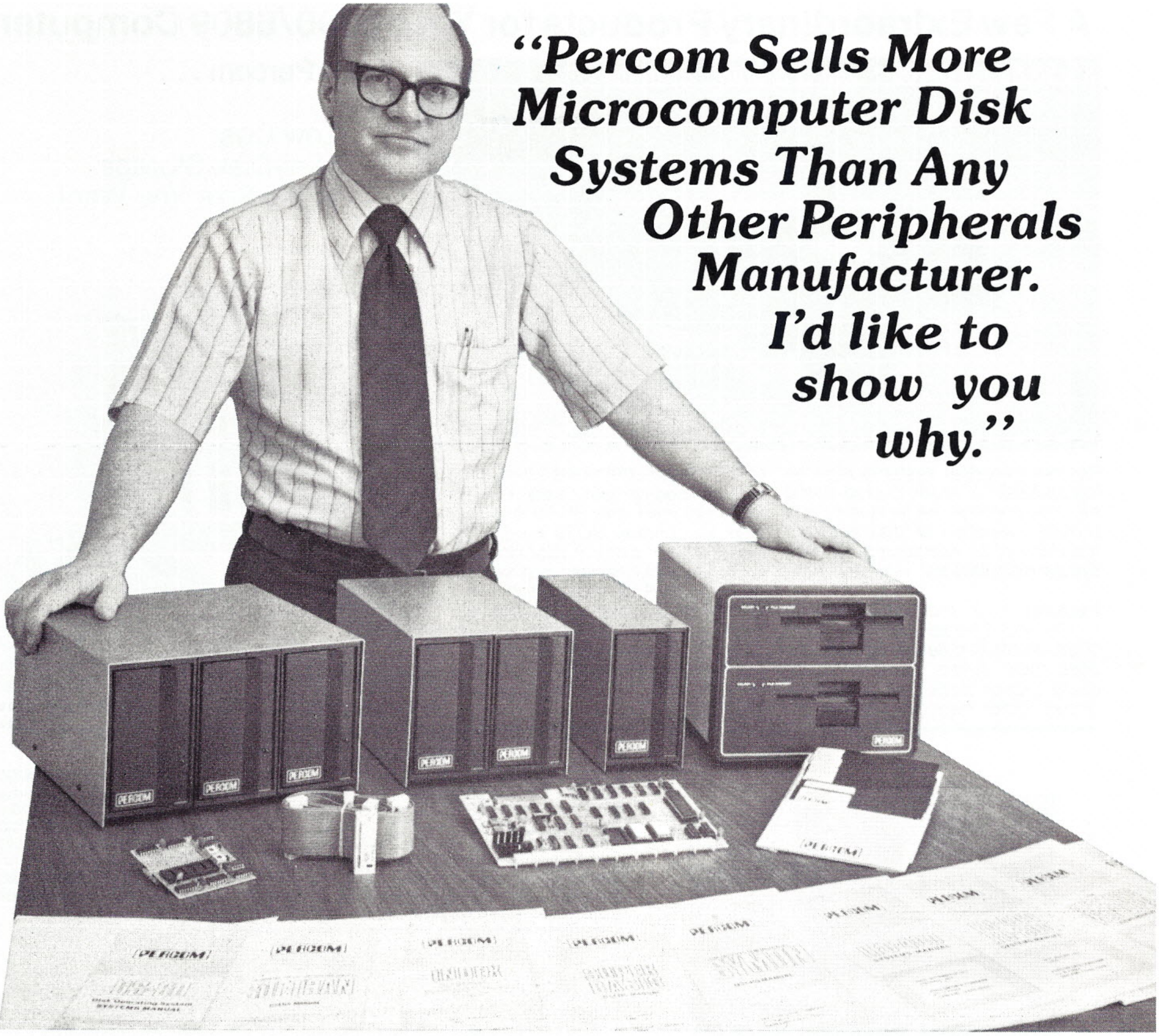
Workbook 6—PET Control and Logic Statements describes testing and branching, subroutine use and logic operations. It provides more overall information on how to write a BASIC program.

Please address all correspondence directly to Robert Baker, 15 Windsor Drive, Atco NJ 08004.

CONTEST

The winner of the $500 prize in our "best article of the year" contest is R. M. Law and D. C. Mitchell's "A Text Formatter in BASIC" from the May 1979 issue. Many thanks to all who voted. Our regular "best article of the month" contest has been discontinued. Perhaps we'll do it again in the future. Congratulations to R. M. and D. C.!

"Percom Sells More Microcomputer Disk Systems Than Any Other Peripherals Manufacturer. I'd like to show you why."



"Percom has been manufacturing mini-disk storage systems for microcomputers since 1977 when we introduced the 35-track, single-drive LFD-400™. Now we produce 1-, 2- and 3-drive systems in 40- and 77-track versions, a multi-density MEGABASE™ system and a host of accessories and software.

"Volume not only means experience in critical production and testing operations, it also means we can offer superior design features, extra testing and qualified backup support at very competitive prices.

"I know of no other microcomputer disk system manufacturer who even begins to offer the broad spectrum of disk equipment and programs available from Percom."

"So before you buy a mini-disk system for your 6800, 6809 or TRS-80* computer, take a good look at what the people at Percom have to offer."

Harold Mauch
President, Percom Data Company

Percom disk systems start at only $399.00. Disk systems and other quality Percom products are available at computer dealers nationwide. **Call toll-free, 1-800-527-1592, for the locations of dealers in your area, or to order direct.** ✓17

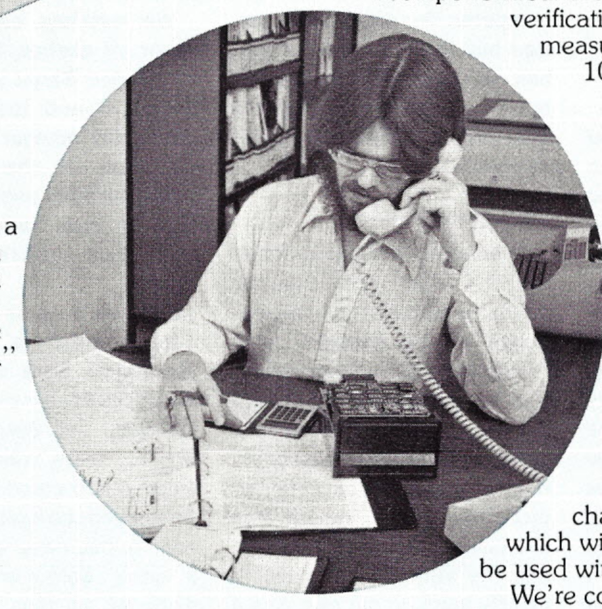
PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.
TM trademark of Percom Data Company, Inc. *trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.

"From an efficient 1K-byte control system DOS to high level languages such as FORTRAN and Pascal, no other microcomputer disk systems manufacturer provides the range and quality of development and application programs available from Percom."

"Connie is running a 'cats eye' test on a mini-disk drive to check radial track alignment. Drive motor-speed timing and sensor alignment tests have already been performed. Disk formatting and format verification tests are next. These measurements are part of the 100% testing every single unit receives."



"Whether you call about a shipping date or ask a tough technical question, you get a competent courteous answer. Outstanding customer service is a hallmark of Percom."

"Richard's making final changes to a disk controller which will allow Percom drives to be used with yet another computer. We're constantly developing and introducing new products that extend and enhance the value of Percom systems."

"Slipping a circuit board through the eye of a needle would be easier than slipping a cold solder joint past Beverly. These are four-drive LFD-400/800 disk system controllers she's inspecting."

Reader Service—see page 258

Broadcasting with a Microcomputer

The folks at WKYB decided to computerize their program log with a TRS-80.

Don Hastings
Box 366
Hemingway SC 29554

I had just invited the computer salesman out of my office never to return until he agreed to speak English, my kind! I felt a computer was what we needed at the radio station, but how was I to select one (and a program) when these guys all insisted on talking RAMs, ROMs, bits, bytes and bauds! Twenty to forty thousand dollars was simply too much money to play around with on something we were not certain would do the job for us. Obviously, if I were to make an intelligent choice, I had to have some idea of just what I was buying. Then I noticed the Radio Shack TRS-80.

We bought it just for our own education and to get some idea of what was involved in using a computer. We knew it was limited but felt it at least would help us understand the basics. Maybe we would even be able to communicate with computer salesmen!

Getting Hooked

The TRS-80 instruction manual is a jewel. With only the ability to read and minimal patience, anyone should be able to operate this computer using that instruction manual. It held our attention better than a best-seller for three days. Best of all, it gave us confidence in approaching that keyboard. The exercises were downright fun, and the author's invitation to modify them on our own was a challenge we could not resist. We wondered why it had all seemed so complicated before. There just isn't anything like hands-on experience.

We watched cannon balls shoot off castles, Snoopy yell at the Red Baron (courtesy of *Kilobaud*) and the computer guess any number we picked. We wrote all the cute little games all other beginners probably do and learned how to compress, abbreviate and cheat.

OK, we became hooked on microcomputers. Even the secretaries were wagering at Blackjack! The mystique of the computer had been lifted and we felt there was little that smart micro couldn't do. Then we learned the limit of 4K memory. We had never experienced a "sorry" before and stared in disbelief at that soon-to-be-hated notation. The prescribed cure was 16K, the limit of the TRS-80 in Level I.

Hard at it viewing a completed log.

Programming the Program Log

With 16K, the world suddenly seemed enormous, and we began considering actually doing a broadcast log on the little giant. A broadcast log is the heart of all daily broadcast operations. It contains every program, newscast, public service announcement and commercial to be aired each day. It must meet strict FCC regulations and thus is prepared with great care. It is called the program log, and just like a computer program, it establishes the order of events for things to happen.

To make a program log, the computer must be able to understand where each item on the log is to go and be able to remember all the information about each commercial account —account number, product code, commercial length, number to be aired that day, what times they can be aired and what priority each account has in choice of air-time. Then the computer must schedule each account only in those spaces allotted for commercials, which means it must remember where these positions are located, what is allowed in each, how much time is available in each position and if a competing account is already occupying part of that space.

Suddenly, 16K and only one array didn't seem like very much. We considered reconsidering. There were so many little decisions we had always taken for granted. The prospect of programming a computer to do it all seemed overwhelming. In short, we panicked.

Then, something from the instruction manual gave us hope . . . the use of subroutines. All those little decisions could be programmed using subroutines! Now, I know most experienced programmers would say, "Dummy, of course you use subroutines." But for us beginners the use of subroutines helped us to see the light at the end of the tunnel.

Subroutines automatically meant we had to break the job down into smaller parts and

tackle each one individually. We no longer were fighting the enormous job of a complete program log, but only a small part of it at a time. Without a doubt, the use of subroutines is what made this job possible in only 16K of memory. Also without a doubt, being able to tackle the overall program in small pieces saved our sanity.

The next best piece of advice the manual gave us was to leave open lines for later additions. Since we were inexperienced, we began leaving 20 lines between statements and 1000 between separate elements of the program. What a blessing that turned out to be!

We wrote our programs on notebook paper before entering them into the computer, but our inexperience showed up quickly when whole routines began crashing because of wrong procedures. Until we learned better, we found it easier to write directly into the computer and run samples as often as possible. Again, there is nothing like hands-on experience to find out what won't work.

We quickly learned to write the display routines first. Not only did it solidify exactly what previous routines must accomplish, it also provided us with a quick visual check of how we were doing. We spotted many errors early because we had the routines written to display the results.

It took us three months, working nights and weekends, to complete the first working program. We fit it into that little Level I 16K TRS-80, which cost less than $1000. It saved the company thousands of dollars by cutting almost in half the time it now takes to do a broadcast log. As an added plus, it provided us with a log that is more consistent day to day since it makes decisions exactly as directed without errors produced by colds, flu, loss of sleep or other foibles in the human makeup. We gained an experience with computers that will be most helpful in making an intelligent decision on future computer needs. There's no doubt that our experiment with a microcomputer has been a success.

We have used the TRS-80 at this station since July 1978 and added a little more to our program, compressed a little more and, with our growth in experience, now have a versatile program doing almost twice as much as the original. Naturally, with Level I and only 16K, there are some limitations.

The Program

In order to obtain a printout, the program listing is a Level II conversion. Originally, for this program to run in 16K, we had to take advantage of every abbreviation allowed in Level I. With the refinements we have added, this factor is not as critical.

Lines 100 to 180 set to zero all arrays that comprise the broadcast day as entered in Line 110. Since we are a day-timer (operating from sunrise to sunset) our broadcast day will vary from month to month. We did not use military time because most of our operators were unfamiliar with it, and we decided facing a computer (even a micro) for the first time was trauma enough. There is no such compromise in the new Level II program being completed.

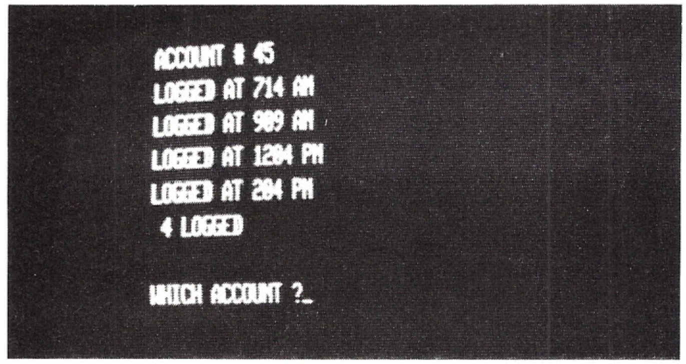
Since we operate on a clear channel, we not only must leave the air between local sunset and sunrise to make way for the clear-channel station, we must also operate at half power in the early morning and evening. Lines 200 and 210 place a special tape to play at the time of power changeover to provide low-level modulation for the change.

We have each hour divided into a maximum of twelve commercial "breaks" with no more than four events to be scheduled in any one break. Thus, each hour has the capability of 48 scheduled events. We spread the twelve breaks evenly around the hour, one every five minutes. It is seldom we use all of these positions, but it provides the capability of placing an event almost precisely when we would like it to air. (See Fig. 1.)

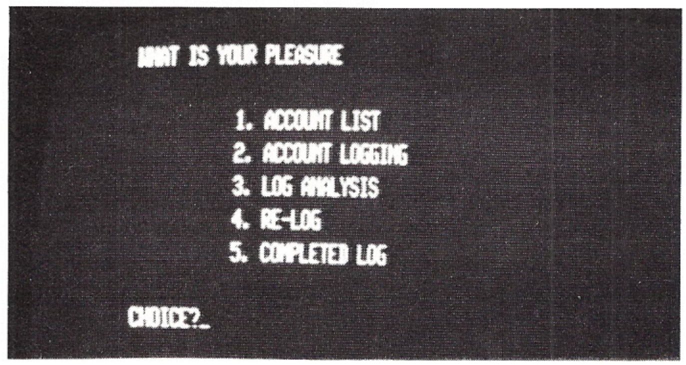
Lines 300 to 370 automatically enter our newscasts into the log positions they will occupy based upon the spread of those positions as described above. Thus, the first newscast position is set at 633 (6 AM, position 33), and we move through the log each hour from there.

Lines 400 to 440 allow us to set aside any portion of an hour or the entire hour for special programs. These might be religious, public affairs, etc., where either no commercial matter is to be allowed or the program has been sold to specific sponsors and the computer is being told to keep its cotton-picking hands off that time!

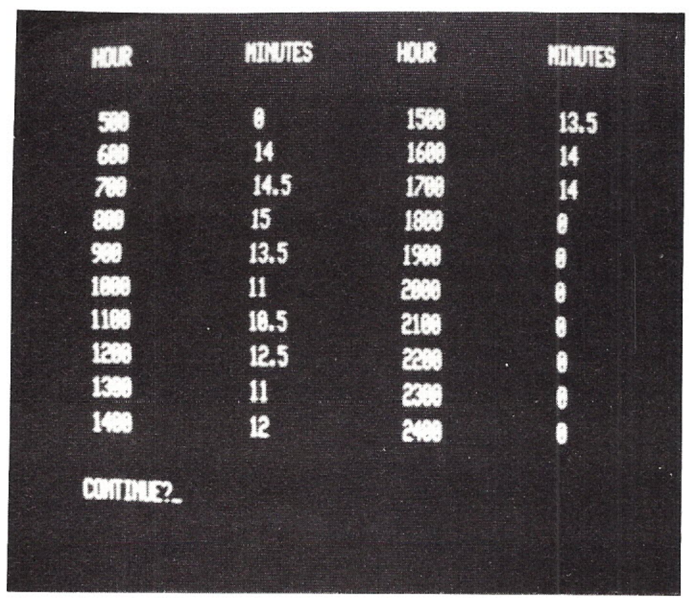
Lines 600 to 680 schedule special announcement and jingle tapes, which play during periods when we are completely



The computer displays the spread of an individual account.



Selection of displays and additional routines after completing a log.



Commercial minutes logged each hour in a completed log.

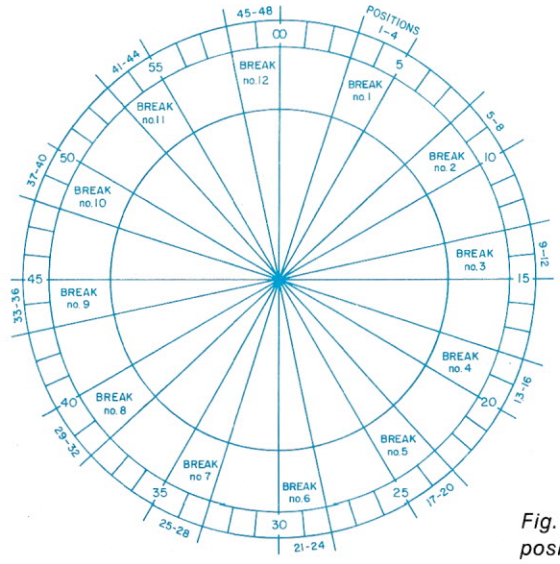


Fig. 1. Provision for 12 "breaks" per hour, four positions per break.

automated on the air. Lines 800 to 900 add weather and headline sponsors to the log, and add their commercial time to the totals for each hour.

Lines 1000 to 1220 provide the capability of logging an event either at a precise time on the log or within a specified time period. This routine even has a nice little display to inform the operator there are no availabilities in the time periods requested. We debated on including that display for fear the operator would beat the machine to death.

Lines 2000 to 2070 are the entry points for all other accounts not sponsoring special programs or specifying time periods for their commercials. These accounts will be spread throughout the broadcast day.

Lines 3000 to 4360 are the workhorse of the whole program. This routine makes the decisions what to do with all the commercials that have been fed into the computer's general accounts. Things considered during the logging process include the spread of the commercials throughout the day, product

Program listing.

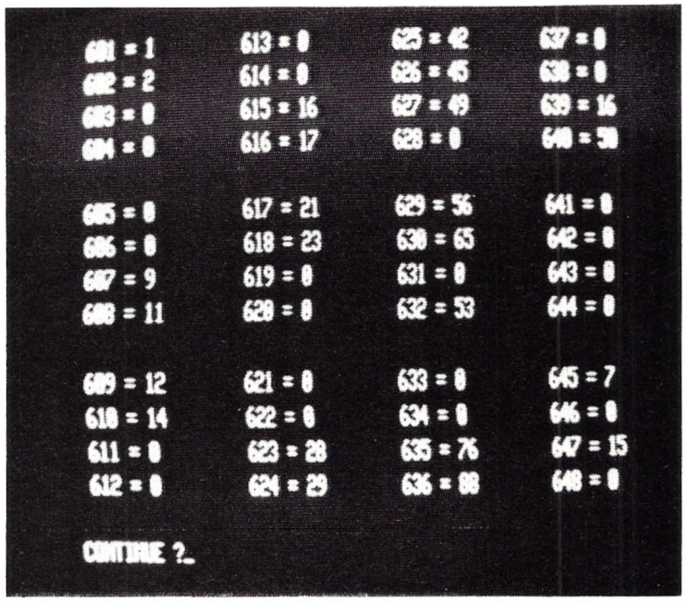
```
10 REM ** WRITTEN BY DON HASTINGS, 6/78
11 REM ** LAST UPDATE 12/78
20 X=500
30 A(X)=0:X=X+100:IF X<2100 GOTO30
100 CLS:PRINTTAB(18),"S T O P   T A P E":Y=1:N=0
110 PRINT"SET LOG":GOSUB9000:F=U:G=V:PRINT"CLEARING":X=1
120 FORL=D TO D+48:A(L)=0:NEXTL:D=D+100:IF D<=E GOTO120
150 FORA=X TO X+94:A(A)=0:NEXTA:X=X+100:IF X<200 GOTO150
160 FORA=91TO95:A(A+100)=5:NEXTA:A(100)=0:A(200)=0:X=549
170 FORC=X TO (X+24):A(C)=0:NEXTC
180 X=X+100:IF X<900 GOTO170
200 CLS:PRINT"POWER CHANGE"
210 INPUT"AM ";X:A(X)=95:INPUT"PM ";X:A(X+1200)=95
300 CLS:PRINT"LOGGING NEWS":PRINT:X=633
310 FORL=X TO (X+7):IF L+7>G GOTO350
320 IF L<F X=X+100:GOTO310
330 A(L)=100:NEXTL:M=INT(X/100)*100:A(M)=120:IFX<1033 X=X+100:GOTO310
340 X=X+200:GOTO310
350 INPUT"ENTER UNSPONSORED ";L:IF L=0 GOTO410
360 INPUT"AM ";X:IF X=0 L=L+1200
365 M=INT(L/100)*100:A(M)=0
370 L=M+33:A(L)=0:A(L+1)=0:A(L+2)=0:GOTO350
410 CLS:PRINT"PROGRAMS":GOSUB9000:IF U=0 GOTO500
420 FORL=D+1 TO D+48:IF(L>=U)*(L<=V+1) A(L)=100
430 NEXTL:D=D+100:IF D<V GOTO420
440 GOTO410
500 A(849)=91:A(850)=92:A(851)=93:A(852)=94:A(649)=95:A(549)=96
610 CLS:PRINT"AUTOMATION":GOSUB9000:IF U=0 GOTO800
620 L=D+8:GOSUB700:L=D+32:GOSUB700
625 L=D+12:GOSUB700:L=D+24:GOSUB700
630 L=D+36:GOSUB700:L=D+40:GOSUB700
640 L=D+20:GOSUB730:L=D+44:GOSUB730
650 L=D+4:GOSUB750:L=D+28:GOSUB750
660 L=D+16:IF (L>U)*(L<V)*(A(L)=0) A(L)=92
670 D=D+100:IF L+8<V GOTO620
680 GOTO610
700 IF (L>U)*(L<V)*(A(L+4)<>100)*(A(L)=0) A(L)=93
710 RETURN
730 IF (L>U)*(L<V)*(A(L)=0) A(L)=94
740 RETURN
750 IF (L>U)*(L<V)*(A(L)=0) A(L)=91
760 RETURN
800 X=1:A=90:Q=853
810 CLS:PRINT"WEATHER FLIGHT ";X:GOSUB9200:IF A(A)=0 GOTO920
850 ON X GOTO860,870,880,890,900,910,920
860 L=704:T=1300:GOSUB960:GOTO810
870 L=728:T=1300:GOSUB960:GOTO810
880 L=1304:T=1800:GOSUB960:GOTO810
890 L=1328:T=1800:GOSUB960:GOTO940
900 L=716:T=1300:GOSUB960:GOTO940
910 L=1316:T=1800:GOSUB960:GOTO1000
920 X=X+1:IF X<5 GOTO810
930 IF X>6 GOTO1000
940 CLS:PRINT"HEADLINE FLIGHT ";X-4:GOSUB9200:IF A(A)=0 GOTO920
950 GOTO850
960 IF L<F L=L+100
970 IF(L>T)+(L>G) X=X+1:A(Q)=A:Q=Q+1:A=A-1:RETURN
980 IF A(L)<>100 A(L)=A
990 M=INT(L/100)*100:A(M)=A(M)+A(A+100):L=L+100:GOTO960
1000 Q=859
1001 CLS:PRINT"SPECIAL LOGGING":GOSUB9200:IF A(A)=0 GOTO1010
1002 INPUT"HOUR ";U:INPUT"AM ";X:IF X=0 U=U+1200
1003 INPUT"POSITION ";L:INPUT"CORRECT ";X:IF X=0 GOTO1002
1004 A(U)=A(U)+A(A+100):L=L+U:A(Q)=A:A(L)=A:Q=Q+1:A=A-1:GOTO1001
1010 RESTORE:T=0:CLS:PRINT"SPECIFIED TIMES":PRINT:PRINT"ACCOUNT #";A
1020 INPUT"CODE .... ";A(A):IF A(A)=0 GOTO2010
1030 INPUT"LENGTH .. ";A(A+100):INPUT"SPOTS .... ";A(A+200)
1035 PRINT:INPUT"CORRECT ";X:IF X=0 GOTO1010
1040 X=A(A):GOSUB9300
1050 IF A(Q)<>0 Q=Q+1:GOTO1050
1055 A(Q)=A
1060 CLS:PRINT"ACCOUNT #";A;" - SPOTS LEFT  ";A(A+200):GOSUB9000
1070 PRINT:INPUT"SPOTS .... ";W:IF W>A(A+200) GOTO1060
1080 READ S:L=S+D+M:T=T+1:IF T=13 GOTO1160
1085 IF L<U L=L+100
1090 IF(A(L)<>0)*(L<V) L=L+100:GOTO1090
1100 Z=A(A+100):IF L>V GOTO1080
1105 IF M=0 X=A(L+3):Z=Z+A(X+100):IF A(X)=A(A) GOTO1150
1110 IF M<=1 X=A(L+2):Z=Z+A(X+100):IF A(X)=A(A) GOTO1150
1115 IF M<=2 X=A(L+1):Z=Z+A(X+100):IF A(X)=A(A) GOTO1150
1117 IF M=2 X=A(L-2):Z=Z+A(X+100)
1118 IF M>=1 X=A(L-1):Z=Z+A(X+100)
1119 IF Z>92 GOTO1150
1120 C=INT(L/100)*100:A(C)=A(C)+A(A+100)
1125 A(L)=A:W=W-1:A(A+200)=A(A+200)-1:IF A(A+200)=0 GOTO1190
1130 IF W<>0 GOTO1150
1140 CLS:RESTORE:T=0:GOTO1060
1150 L=L+100:GOTO1090
1160 PRINT:PRINT"UNABLE TO LOG";W:INPUT"OTHER TIMES AVAILABLE ";X
1180 RESTORE:T=0:IF X=1 GOTO1060
1190 A=A-1:GOTO1010
1200 IF A(X)=A(A) L=L+100:GOTO1090
1210 Z=Z+A(X+100):IF Z>92 L=L+100:GOTO1090
1220 RETURN
2010 CLS:PRINT"GENERAL ACCOUNTS":A=1:PRINT
2020 PRINT"ACCOUNT #";A:GOSUB9400:CLS:IF A(A+300)<>0 A=A+1:GOTO2020
2025 B=A:A=1:X=1
2030 CLS:PRINT"VERIFICATION":PRINT
2040 PRINT" #    CODE","LENGTH","SPOTS","PRIORITY"
2050 PRINTA;"  ";A(A),A(A+100),A(A+200),A(A+300)
2052 IF A(A)=0 GOTO2060
2055 A=A+1:IF A<X+9 GOTO2050
2060 PRINT:INPUT"CORRECT";Z:IF Z=0 CLS:INPUT"ACCOUNT #";A:GOSUB9400:GOTO2030
2070 X=X+9:IF A(A)<>0 GOTO2030
3000 CLS:PRINT"LOADING":A=1:C=1
3010 X=A(A):IF X<>C GOTO3100
3020 GOSUB9300:M=Q
3030 IF A(M)=0 A(M)=A:GOTO3100
3040 M=M+1:IF M<Q+24 GOTO3030
3050 M=849:Q=M:GOTO3030
3100 A=A+1:IF A<B GOTO3010
3110 A=1:C=C+1:IF C<15 PRINT"CODE",C:GOTO3010
```

separation from competing accounts, maximum time allowed in any one commercial break and maximum commercial time allowed in any one hour.

Finally, we come to the display routines that are selected in lines 5000 to 5080. Because of memory limitations we originally had only the first three displays, but as we learned to condense and improve our program we were able to add three additional display routines. All of these display routines are contained in lines 5090 to 9410 along with the subroutines for account entries and various decision-making elements of the total program.

Older and Less Dumb

Currently we are in the process of completing a completely revised program for Level II. When I look back on the amount of memory consumed in our earlier Level I programs and how we have been able to condense them over the months, it is quite evident how quickly anyone can acquire programming knowledge. At least, I like to think of it as symbolic of our growth, but it really shows how dumb we had to be. That means we aren't necessarily more intelligent now, just less dumb.■



6 AM log page with accounts paged. Numbers 601–648 are log positions, not times.

```
4000 CLS:S=0:P=1:I=1
4010 H=0:J=549
4020 PRINT"LOGGING CAROUSEL";5+H;" PRIORITY";P
4030 FORR=J TO J+24:IF R=J+24 GOTO4340
4035 RESTORE:T=0:K=0:IF A(R)=0 GOTO4335
4040 X=A(R):IF(A(X+200)<1)+(A(X+300)<>P)+(X>B)GOTO4335
4045 C=A(X+200):C=INT(10/C):IF C<1 C=1
4050 L=INT(F/100)*100:K=K+1:IF K>13 GOTO4335
4060 DATA 1,25,13,41,17,45,5,29,21,9,37,33,45
4080 READ V:L=L+V+H:T=T+1:IF T=12 RESTORE:T=0
4090 IF L<F L=L+100:GOTO4090
4100 IF L>G GOTO4050
4105 IF A(L)<>0 L=L+100:GOTO4100
4110 Q=0:Z=A(X):IF L>G GOTO4050
4115 M=INT(L/100)*100:IF A(M)+A(X+100)>1080 GOTO4330
4120 IF A(X+100)=30 GOTO4130
4125 IF A(L+4)=100 L=L+100:GOTO4100
4130 O=1:IF H=3 GOTO4160
4140 U=A(L+O):Q=Q+A(U+100):IF Z=A(U) GOTO4330
4150 O=O+1:IF O+H<4 GOTO4140
4160 O=1:IF H=0 GOTO4190
4170 U=A(L-O):Q=Q+A(U+100):IF Z=A(U) GOTO4330
4180 O=O+1:IF H-O>=0 GOTO4170
4190 IF S=1 GOTO4230
4200 IF Q+A(X+100)<15 GOTO4330
4210 IF Q+A(X+100)>91 GOTO4330
4220 GOTO4300
4230 IF Q+A(A+100)>121 GOTO4330
4300 A(L)=X:A(X+200)=A(X+200)-1:A(M)=A(M)+A(X+100)
4310 PRINT"ACCOUNT";X;" AT";L:IF A(X+200)<=0 GOTO4335
4330 L=L+(C*100):GOTO4100
4335 IF I=0 GOTO6300
4336 NEXTR
4340 H=H+1:J=J+100:IF H<4 GOTO4020
4350 IF S=0 PRINT"VERIFYING":S=1:GOTO4010
4360 IF P<3 S=0:P=P+1:GOTO4010
5010 CLS:PRINT"WHAT IS YOUR PLEASURE"
5020 PRINT:PRINT"1.  ACCOUNT LIST":PRINT"2.  CAROUSEL LOADING"
5030 PRINT"3.  COMPLETED LOG":PRINT"4.  ACCOUNT LOGGING"
5040 PRINT"5.  ATTEMPT RE-LOGGING":PRINT"6.  LOG ANALYSIS"
5070 PRINT:INPUT"CHOICE";X
5080 ON X GOTO5200,8010,7010,5090,6000,5500
5090 PRINT:INPUT"WHICH ACCOUNT ";X:A=0:IF X=0 GOTO5010
5100 CLS:PRINT"ACCOUNT #";X:T=INT(F/100)*100
5110 FORZ=1TO48:IF A(T+Z)=X W=T+(Z/.85):A$="AM":GOTO5140
5120 NEXTZ:T=T+100:IF T<G GOTO5110
5130 PRINTA;"LOGGED":GOTO5090
5140 W=INT(W):IF W>1300 W=W-1200:A$="PM"
5150 PRINT"LOGGED AT ";W;" ";A$:A=A+1:GOTO5120
5200 CLS:PRINT"ACCOUNT     CODE     LENGTH     SPOTS"
5210 FORA=X TO (X+12):PRINTTAB(3),A;TAB(12),A(A);TAB(21),A(A+100);
5220 PRINTTAB(33),A(A+200):IF A(A)<>0 NEXTA
5230 PRINT:INPUT"CONTINUE ";Z:IF(Z=1)*(A(A)<>0) X=X+13:GOTO5200
5240 GOTO5010
5500 CLS:PRINT"HOUR","MINUTES","HOUR","MINUTES"
5510 FORM=500 TO 1200 STEP 100
5520 PRINTM,A(M)/60,M+800,A(M+800)/60:NEXT M
5530 PRINT:INPUT"CONTINUE";X:GOTO5010
6000 CLS:S=1
6010 PRINT:INPUT"ACCOUNT #";A:J=549:IF A=0 GOTO5010
6030 FORR=J TO J+23:IF A(R)=A A(R)=0:GOTO6100
6040 NEXTR:J=J+100:IF J<850 GOTO6030
6050 INPUT"UNABLE TO LOCATE - WILL YOU ADD ";X:IF X=0 GOTO6010
6060 GOSUB9400:B=A+1:GOTO6110
6100 PRINT"LOCATED IN CAROUSEL ";INT(R/100)
6110 INPUT"MOVE TO ";J:J=(J*100)+49:IF J=0 GOTO6010
6120 FORR=J TO J+23:IF A(R)=0 A(R)=A:GOTO6200
6130 NEXTR:PRINT:PRINT"UNABLE":GOTO6110
6200 CLS:PRINT"LOG ADJUSTMENT":X=INT(F/100)*100:P=A(A+300)
6210 FORL=X+1 TO X+48:IF A(L)=A A(L)=0:A(A+200)=A(A+200)+1:GOTO6240
6220 NEXTL:X=X+100:IF X<G GOTO6210
6230 I=0:H=INT(J/100)-5:PRINT"LOGGING";A(A+200),"SPOTS":GOTO4035
6240 M=INT(L/100)*100:A(M)=A(M)-A(A+100):GOTO6220
6300 PRINTA(A+200);" REMAINING":GOTO6010
7010 PRINT:INPUT"HOUR ";T:T=INT(T/100)*100
7020 INPUT"AM ";X:CLS:IF X=0 T=T+1200
7030 U=1:V=13:W=25:Z=37
7040 PRINTT+U;"=";A(T+U),T+V;"=";A(T+V),T+W;"=";A(T+W),T+Z;"=";A(T+Z)
7050 U=U+1:V=V+1:W=W+1:Z=Z+1:IF(U=5)+(U=9)PRINT
7055 IF U<13 GOTO7040
7060 PRINT:INPUT"CONTINUE ";X:CLS:IF X=0 GOTO5010
7070 T=T+100:IF T>G GOTO5010
7080 T=INT(T/100)*100:GOTO7030
7090 GOTO7040
8010 X=5:R=549:S=1
8020 CLS:PRINT"   C A R O U S E L   ";X:PRINT
8030 PRINTS;" = ";A(R),S+12;" = ";A(R+12)
8040 S=S+1:R=R+1:IF S=13 GOTO8060
8050 GOTO8030
8060 R=R-12:R=R+100:S=1:X=X+1
8070 PRINT:INPUT"CONTINUE ";Z:IF(Z=1)*(X<9)GOTO8020
8080 GOTO5010
9000 PRINT:INPUT"ENTER BEGINNING TIME: ";U:IF U=0 RETURN
9010 INPUT"IS TIME AM ";X:IF X=0 U=U+1200
9020 INPUT"ENTER ENDING TIME: ";V
9030 INPUT"IS TIME AM ";X:IF X=0 V=V+1200
9040 D=INT(U/100)*100:U=(D+1)+(U-D)*.8
9050 E=INT(V/100)*100:V=E+(V-E)*.8
9060 PRINT:PRINT"FIRST POSITION = ";U:PRINT"LAST POSITION =  ";V
9070 PRINT:INPUT"CORRECT ";X:IF X=0 CLS:GOTO9000
9080 RETURN
9200 PRINT:PRINT"ACCOUNT #";A:INPUT"CODE .... ";A(A):IF A(A)=0 RETURN
9210 INPUT"LENGTH .. ";A(A+100):PRINT:INPUT"CORRECT ";Z:IF Z=0 CLS:GOTO9200
9220 RETURN
9300 Q=749:M=2
9310 IF(X=2)+(X=3)+(X=5)+(X=10) Q=649:M=1
9320 IF(X=1)+(X=13) Q=549:M=0
9330 RETURN
9400 PRINT:INPUT"CODE ......  ";A(A):INPUT"LENGTH ....  ";A(A+100)
9410 INPUT"SPOTS .....  ";A(A+200):INPUT"PRIORITY ..  ";A(A+300):RETURN
```

Automatic Selection of Plotting Limits

Program that determines optimum limits for graphics and charts.

William D. Johnston
1808 Pomona Drive
Las Cruces NM 88001

The presentation of data in graphic form has many advantages over the tabular printouts that have been standard to the computer field for many years. Plots show trends in data and give an overall picture of the process or event being graphed. In most cases, the plot is not intended to replace the tabular listing; rather, it is a useful supplement that permits the user to observe patterns, tendencies, changes and so forth. To a great extent this frees the analyst from the tedious and time-consuming chore of examining large lists of numerical data.

Nevertheless, if we are to profit from its benefits, the graphical presentation must in some way be referenced to the data. The usual way we do this is by numerically annotating the axes of the plot in a manner that will enable us to tell at a glance the approximate value of the data at any point. We don't need to know the *exact* value; we can always go back to the printout for that. But we do need a reasonable system of reference.

For example, we might have a plot whose x-axis is ten inches long, representing, say, an elapsed time from 0 to 100 seconds. Logically, we would label the axis from 0 to 100 in increments of ten seconds (i.e., 0, 10, 20, 30 . . . 100). See Fig. 1.

But what if we were only interested in the data from 71 to 88 seconds? We could plot the data using the same scale of annotation (0 to 100), but in that case most of the plot would be blank, while the data we were interested in would be crammed into less than two inches of length near the end.

Clearly, the best solution is to label one end of the axis 70, the other end 90 and the intermediate points (i.e., the one-inch intervals) in steps of 2. The plot annotation points would then appear as 70, 72, 74 . . . 90 (Fig. 2). This expanded plot would fill most of the page and would still be easy to read.

The process must be repeated every time a plot is produced if we are to always optimize the presentation. The job becomes more difficult, however, when the raw limits that we start out with are of values for which optimum round limits are not easily estimated by "eyeball" techniques. As the total number of plots increases, the burden of selecting appropriate limits increases correspondingly.

This was the situation that confronted a large-scale data processing operation for which I designed and implemented a production graphics system some years ago. At its peak the system generates in excess of 30,000 plots per month (more than a thousand plots a day, seven days a week) and 4000 per month during slack periods. Obviously, it wasn't practical to hire a whole crew of people just to eyeball data and select plotting limits.

Consequently, I incorporated into the system a small subroutine whose funciton is to examine the raw minimum and maximum values of the data for each axis. Then, based on the given lengths of the axes, it returns optimized round limits and increments. The plots are thereby spread out over as much of the page as possible, and the numerical annotation is easy to interpret.

Other parts of the graphics software package write the numerical annotation (as well as all titles and other written legends) directly onto the plots. These graphs are in every respect complete the moment they come off the plotter, and they are inserted directly into printed reports.

Few computer operations have a volume of graphics production that anywhere near approaches that of the system described above, but even the smallest hobby and business systems can make use of its features. Of course, fully optimized plotting limits may not be suitable in every instance; it may be desirable in some cases to sacrifice optimization for standardization. These few cases notwithstanding, the user frequently finds it valuable to generate a set of fully optimized test plots before deciding on a standard layout.

The Round Limits Subroutine

Listing 1 is a BASIC version of the Round Limits subroutine used in the production graphics system described above. (The original program was a FORTRAN subroutine written in 1969, which, in turn, was a modified version of an assembly-language routine that I wrote in 1966.) The BASIC program is designed as a subroutine and is extremely easy to use. It consists of only about 60 executable statements and requires only a few milliseconds of execution time.

Full instructions on the use of the subroutine are given by the remarks contained within the program, but briefly the procedure is as follows: The raw minimum and maximum values, as well as the length of the axis, are supplied by the main program. This is done by setting A1, B1 and L, respectively, equal to those values. A GOSUB 8000 is executed, and upon return from the subroutine the round minimum, round maximum and round increment will be found in variables A, B and C, respectively. The foregoing procedure is repeated for each axis of the plot for which round limits are desired.

Note that for the purposes of this subroutine the words "minimum" and "maximum" simply refer to the numerical values at the two extremities of a given axis. There are, in fact, no restrictions on their values, and either may be algebraically greater or less than the other and may be either positive or negative.

The method by which you initially obtain the raw minimum and maximum values is determined by the nature of the operation that is generating the data to be plotted. If, for example, the data is being written onto a file for later use by the graphics software, it is a simple matter to detect the minimum and maximum values of each parameter as it is generated or written out. The actual method must suit the circumstances, of course, but in any case it is a process that must be completed before the Round Limits routine is called.

The length of the axis may be in any units, where the units of length refer to the intervals at which the numerical annotation

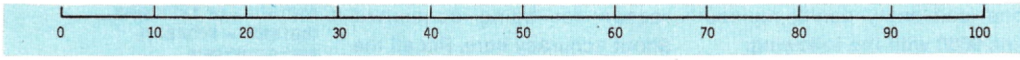


Fig. 1. If an axis of a plot is ten units long and is to represent an elapsed time from 0 to 100 seconds, it is a simple matter to annotate the axis in a manner that is easy to read.

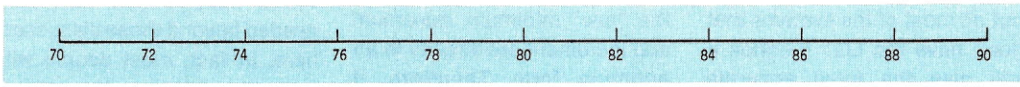


Fig. 2. Suppose that even though all of the same data as described in Fig. 1 is available to us, we are actually interested in displaying only the portion that lies between 71 and 88 seconds. Assuming that we use the same length axis, the most readable presentation is obtained by annotating the axis from 70 to 90 in increments of 2.

is to appear. For example, plots are frequently drawn on a grid with one-inch spacing between major divisions, and the numerical annotation is placed at these same intervals. In such cases you would simply specify the axis length in inches.

Another common situation is to generate a plot on a video display where the grid has no direct relationship to conventional units such as inches or centimeters. (Indeed, the underlying software is in all probability concerned with pixels and bit patterns, rather than lengths.) Suppose that a grid on such a display is divided into five equal intervals, and that the numerical annotation is to appear at each of these intervals. These arbitrary (but equal) intervals become our units of measure, so the length therefore has a value of 5.

Though it is customary to arrange plot axes so that their lengths are integral multiples of the basic units of measurement, the Round Limits subroutine does not require that this be done. For example, you could declare an axis length of 6.3 inches. The returned round values, A and B, would correspond to the two extremities of the axis (i.e., 0 and 6.3 inches), and the value C would be the round increment between each whole one-inch interval. The value of the axis at the far end (at 6.3 inches) would not be a round number (unless by coincidence), but the annotation at every whole interval from 0 through 6 inches *will* be round.

Specific Implementation Considerations

The key to a successful round limits routine lies not in the selection of the minimum and maximum values, but in the selection of the appropriate increment per unit of axis length. An examination of Listing 1 will disclose that the increment is computed first, and then the minimum and maximum values are allowed to fall into place on the basis of the computed round increment and the given axis length. The value of the raw increment is factored into a fractional part (called the mantissa) and a power of ten (the characteristic). The mantissa is rounded up to some convenient standard value and then recombined with the characteristic to arrive at the computed round increment.

The decision as to just what is a "convenient standard value" for the mantissa is purely subjective. Most people would agree that a mantissa of 1, yielding increments of 1, 10, 100, 1000, etc. (or .1, .01, .001, etc.), is convenient. The same is true for a mantissa of 5 (for increments of 5, 50, 500, etc., .5, .05, .005, etc.). But what about a mantissa of 6, which yields increments of 6, 60, 600, etc. (or .6, .06, .006, etc.)? Is an axis labeled from 120 to 540 in steps of 60 (see Fig. 3) satisfactory for your purposes?

This Round Limits routine contains a table of standard increment mantissas (lines 8690 and 8700) that may be changed to suit your needs. As given in Listing 1, the table contains 12 different values, and that is the form that has been satisfactory for most of my applications.

Having a large number of entries in the table ensures that the data will be spread out as much as possible along each axis of the plot. On the other hand, it introduces mantissas that may not be visually appealing on your plots. If this is the case for you, change the table so that it includes only the values 1.0, 2.0, 2.5, 5.0 and 10.0. You will also need to change both the dimension of S and the value of M to 5 (to correspond to the number of entries in the table) at lines 8530 and 8560.

You can actually use any (and as many) values you care to in the table. The only restrictions are that the values must be positive, must be listed in order of increasing magnitude, and the last entry in the table must be equal to exactly ten times the value of the first entry. As pointed out above, the dimension of S and the value of M must be equal to the number of entries in the table.

At lines 9150, 9280 and 9450 the ** operator has been used to signify exponentiation. If your interpreter does not accept the double asterisk, replace it with

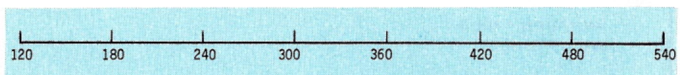


Fig. 3. The use of certain values in the table of standard increment mantissas may produce plot annotation values that are not appealing to all people. In this example, a seven-unit axis is labeled from 120 to 540 in steps of 60. Some people may find this perfectly readable, while others may not like it at all. The values within the table may be changed to suit your needs.

the appropriate symbol (usually ↑) in these three lines.

A base 10 logarithm is computed in line 9080. The interpreter that I use has a base 10 log function (called LGT), so I made use of it in order to obtain the greatest possible accuracy in the result. If your set of library functions doesn't include this one (and many don't), replace line 9080 with the following:

```
9080 LET P = LOG(C2)/LOG(10.0)
```

This is not the most efficient way to compute a base 10 log, but on most of the systems that don't have the LGT function it will give the most accurate results.

Incidentally, there is a good reason for being concerned about accuracy here. Recall the earlier example where data running from 0 to 100 is plotted along a 10-inch axis. In this case the "raw" minimum, maximum and increment are already in an optimum form. Therefore, if these values are supplied to the Round Limits routine, it should return exactly the same values. If the log isn't computed accurately—for example, if you get 1.9999 . . . instead of 2 for $\log_{10}(100)$—the routine will compute new round values with wider separation between the minimum and maximum values.

The SGN function, used at lines 9110 and 9580, is a standard item on almost all BASIC interpreters. For those few who do not have it, the following changes need to be implemented. First, delete line 9110 and replace it with the following three lines:

```
9101 LET N = INT(ABS(P))
9102 GO TO 9150
9110 LET N = INT(P)
```

Then delete line 9580 and insert the following four lines:

```
9580 IF I> = 0.0 THEN 9583
9581 LET I = -INT(ABS(I))
9582 GO TO 9610
9583 LET I = INT(I)
```

There should be no further system-dependent changes needed beyond those discussed here. In fact, most people will probably need to make only the changes regarding the exponentiation operator and the log function. You can save a considerable amount of storage space by deleting the remark statements when you implement the subroutine, but if you do so, keep a copy of this article, including Listing 1, on file for future reference.

Figs. 4a through 4f show some typical examples of use of the Round Limits routine. In each case a raw minimum, raw maximum and an axis length are listed. If these values are supplied to the Round Limits subroutine, it will return the optimized round limits and increment. A sample annotated axis is illustrated to show what the figures would look like on a finished plot.

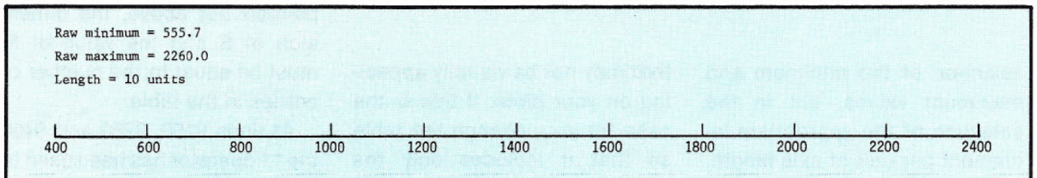


Fig. 4a.

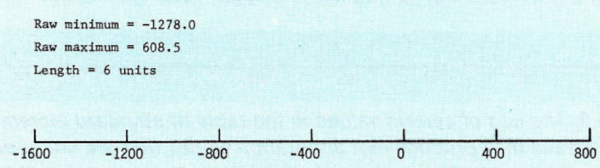


Fig. 4b.

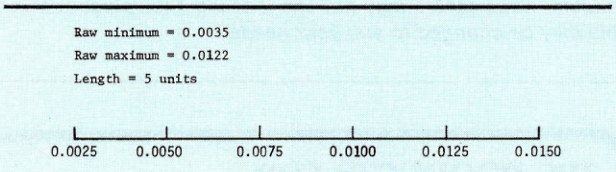


Fig. 4c.

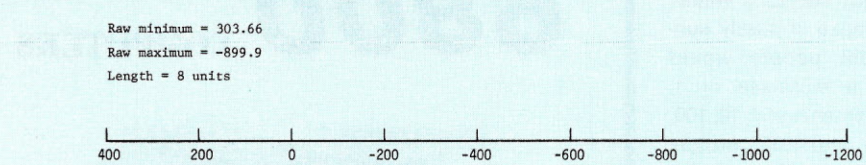


Fig. 4d.

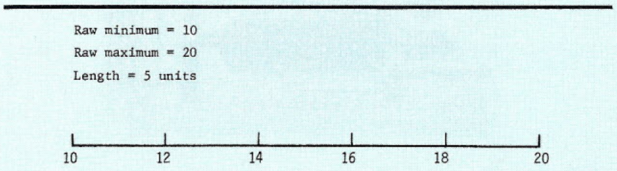


Fig. 4e.

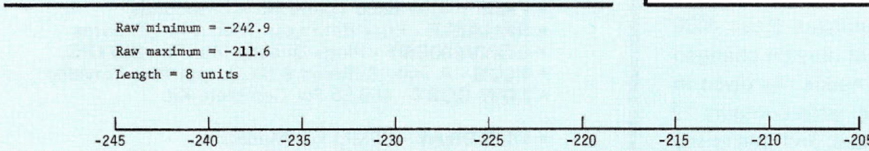


Fig. 4f.

Figs. 4a–4f. Several examples are shown to illustrate the results that can be obtained with the Round Limits routine. In each case a raw minimum, raw maximum and an axis length are listed. If these figures are supplied to the Round Limits routine, it will return the optimized round limits and increment. The sample annotated axis in each illustration shows how these round figures would appear on a finished plot. Compare these samples to the appearance a plot would have if the raw minimum and maximum values were used directly.

Summary

The use of computer graphics is becoming increasingly widespread as more and more businesses and hobbyists recognize its advantages. The ideal automated graphics system permits the user to process and display large quantities of data with a minimum of operator intervention. Those features that increase the ease of interpretation of data, while simultaneously decreasing the user's workload, are important elements in a complete graphics system.

The implementation of the Round Limits routine represents a significant step toward achievement of that goal with only a negligible expenditure in program storage and execution time. No system is too small to profit from its benefits. ■

Reference

William, D. Johnston, "Computer Graphics in a Production Environment," Transactions of the Twenty-Third Conference of Mathematicians, NASA-Langley Research Center, May 11, 1977.

Listing 1. Round Limits subroutine.

```
8000 REM      ROUND LIMITS ROUTINE FOR GRAPHICS.
8010 REM
8020 REM      WILLIAM D. JOHNSTON  --  APRIL, 1979
8030 REM
8040 REM
8050 REM      THIS ROUTINE GENERATES OPTIMIZED ROUND LIMITS
8060 REM      AND INCREMEMTS FOR PLOTTING PURPOSES, BASED ON
8070 REM      GIVEN RAW MINIMUM AND MAXIMUM VALUES AND A
8080 REM      PREDETERMINED AXIS LENGTH.
8090 REM
8100 REM      FOR THE PURPOSES OF THIS ROUTINE, 'MINIMUM'
8110 REM      AND 'MAXIMUM' SIMPLY REFER TO THE NUMERICAL
8120 REM      VALUES AT THE TWO EXTREMITIES OF A GIVEN AXIS.
8130 REM      THERE ARE NO RESTRICTIONS ON THEIR VALUES, AND
8140 REM      EITHER MAY BE ALGEBRAICLY GREATER OR LESS THAN
8150 REM      THE OTHER.
8160 REM
8170 REM      THE ROUTINE IS DESIGNED FOR APPLICATIONS WHERE
8180 REM      THE NUMERICAL ANNOTATION ON THE PLOT IS TO BE
8190 REM      PLACED AT INTEGRAL MULTIPLES OF THE BASIC
8200 REM      UNITS OF AXIS LENGTH (INCHES, CENTIMETERS,
8210 REM      ETC.).
8220 REM
8230 REM
8240 REM
8250 REM      THE FOLLOWING PARAMETERS MUST BE DEFINED
8260 REM      BEFORE THIS SUBROUTINE IS CALLED:
8270 REM
8280 REM          A1   IS THE RAW GIVEN 'MINIMUM' SUPPLIED BY
8290 REM               THE CALLING ROUTINE.
8300 REM
8310 REM          B1   IS THE RAW GIVEN 'MAXIMUM' SUPPLIED BY
8320 REM               THE CALLING ROUTINE.
8330 REM
8340 REM          L    IS THE GIVEN LENGTH, IN ANY UNITS OF
8350 REM               MEASURE, SUPPLIED BY THE CALLING
8360 REM               ROUTINE.
8370 REM
8380 REM
8390 REM      THE FOLLOWING PARAMETERS ARE COMPUTED BY THIS
8400 REM      SUBROUTINE:
8410 REM
8420 REM          A    IS THE COMPUTED ROUND 'MINIMUM' VALUE.
8430 REM
8440 REM          B    IS THE COMPUTED ROUND 'MAXIMUM' VALUE.
8450 REM
8460 REM          C    IS THE COMPUTED ROUND INCREMENT (PER
8470 REM               UNIT OF AXIS LENGTH).
8480 REM
8490 REM
8500 REM
8510 REM
8520 REM
8530     DIM  S(12)
8540 REM      THE VALUE OF M MUST BE EQUAL TO THE DIMENSION
8550 REM      OF S.
8560     LET M = 12
8570 REM      THE TABLE OF STANDARD INCREMENT MANTISSAS IS
8580 REM      READ INTO THE ARRAY S.  IT MAY CONTAIN ANY
8590 REM      NUMBER OF STANDARD INCREMENTS (AS LONG AS S IS
8600 REM      DIMENSIONED BY THAT SAME NUMBER), AND MAY
8610 REM      CONTAIN ANY DESIRED VALUES FOR THESE INCRE-
8620 REM      MENTS.  HOWEVER, THE INCREMENTS IN THE TABLE
8630 REM      MUST BE LISTED IN ORDER OF INCREASING VALUE,
8640 REM      AND THE LAST ENTRY IN THE TABLE MUST BE EQUAL
8650 REM      TO TEN TIMES THE VALUE OF THE FIRST ENTRY IN
8660 REM      THE TABLE.
8670     FOR I = 1 TO 12
8680     READ  S(I)
8690     DATA  1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 6.0, 7.0
8700     DATA  8.0, 9.0, 10.0
8710     NEXT I
8720     RESTORE
8730 REM      SAVE THE GIVEN MINIMUM AND MAXIMUM VALUES.
8740     LET A2 = A1
8750     LET B2 = B1
8760 REM
8770 REM      COMPUTE THE RANGE BETWEEN THE GIVEN MINIMUM
8780 REM      AND GIVEN MAXIMUM VALUES.  THE SIGN OF THE
8790 REM      COMPUTED RANGE WILL INDICATE DIRECTION.
8800 REM      (POSITIVE INDICATES THAT A1 IS LESS THAN B1.
8810 REM      NEGATIVE INDICATES THE OPPOSITE).
8820 REM
8830     LET R = B2 - A2
8840 REM
8850 REM      IF THE GIVEN MINIMUM AND MAXIMUM VALUES WERE
8860 REM      EQUAL TO EACH OTHER, OR IF THE GIVEN AXIS
8870 REM      LENGTH WAS NOT GREATER THAN ZERO, THEN SET
8880 REM      ALL COMPUTED VALUES TO ZERO AND RETURN TO THE
8890 REM      CALLING PROGRAM, AS NOTHING CAN BE DONE.
8900 REM
8910     IF  R = 0.0   THEN 8930
8920     IF  L > 0.0   THEN 9030
8930     LET A = 0.0
8940     LET B = 0.0
8950     LET C = 0.0
8960     RETURN
8970 REM
8980 REM      COMPUTE THE ABSOLUTE VALUE OF THE GIVEN INCRE-
8990 REM      MENT.  THIS IS THE RAW INCREMENT AS COMPUTED
9000 REM      FROM THE GIVEN MINIMUM AND MAXIMUM VALUES, AND
9010 REM      THE LENGTH OF THE AXIS.
9020 REM
9030     LET C2 = ABS(R/L)
```

```
9040 REM
9050 REM      COMPUTE THE POWER (THE CHARACTERISTIC) OF THE
9060 REM      GIVEN INCREMENT.
9070 REM
9080      LET P = LGT(C2)
9090      IF  P >= 0.0  THEN 9110
9100      LET P = P - 1.0
9110      LET N = SGN(P) * INT(ABS(P))
9120 REM
9130 REM      COMPUTE THE MANTISSA OF THE GIVEN INCREMENT.
9140 REM
9150      LET F = C2 / 10.0**N
9160 REM
9170 REM      FIND THE FIRST STANDARD INCREMENT MANTISSA
9180 REM      IN THE TABLE WHICH IS EQUAL TO OR GREATER
9190 REM      THAN THE COMPUTED RAW MANTISSA.
9200 REM
9210      FOR J = 1 TO M
9220      IF  F > S(J)  THEN 9310
9230 REM
9240 REM      COMBINE THE TABLE VALUE MANTISSA AND THE
9250 REM      COMPUTED CHARACTERISTIC TO GET THE
9260 REM      COMPUTED ROUND INCREMENT.
9270 REM
9280      LET C = S(J) * (10.0**N)
9290      LET K = J
9300      GO TO 9570
9310      NEXT J
9320 REM
9330      PRINT  'ABORT --  MOST PROBABLE CAUSE IS AN ERROR'
9340      PRINT  '          IN THE TABLE OF STANDARD'
9350      PRINT  '          INCREMENT MANTISSAS.'
9360      STOP
9370 REM
9380 REM
9390      LET K = K + 1
9400 REM
9410 REM      COMBINE THE TABLE VALUE MANTISSA AND THE
9420 REM      COMPUTED CHARACTERISTIC TO GET THE COMPUTED
9430 REM      ROUND INCREMENT.
9440 REM
9450      LET C = S(K) * (10.0**N)
9460 REM
9470 REM      THE STATEMENTS FROM HERE THROUGH LINE NUMBER
9480 REM      9740 SELECT A NEW MINIMUM VALUE WHICH IS AN
9490 REM      INTEGRAL MULTIPLE OF THE ROUND INCREMENT, AND
9500 REM      IS THE FIRST SUCH VALUE EQUAL TO OR LESS THAN
9510 REM      THE GIVEN MINIMUM (OR THE FIRST VALUE GREATER
9520 REM      THAN THE GIVEN MINIMUM IF THE GIVEN MINIMUM IS
9530 REM      GREATER THAN THE GIVEN MAXIMUM).  THE SIGN OF
9540 REM      THE INCREMENT IS CHANGED WHEN NECESSARY, TO
9550 REM      SHOW PROPER DIRECTION.
9560 REM
9570      LET I = A2 / C
9580      LET I = SGN(I) * INT(ABS(I))
9590 REM
9600 REM
9610      LET T = I
9620      LET A = T * C
9630      LET D = ABS((A - A2) / R)
9640      IF  A2 < 0.0  THEN 9700
9650      IF  R > 0.0  THEN  9970
9660      IF  D <= 0.0001  THEN 9680
9670      LET A = A + C
9680      LET C = -C
9690      GO TO 9970
9700      IF  R > 0.0  THEN 9730
9710      LET C = -C
9720      GO TO 9970
9730      IF  D <= 0.0001  THEN 9970
9740      LET A = A - C
9750 REM
9760 REM      THE STATEMENTS FROM HERE THROUGH LINE NUMBER
9770 REM      10090 SELECT A NEW MAXIMUM VALUE WHICH IS AN
9780 REM      INTEGRAL MULTIPLE OF THE ROUND INCREMENT, AND
9790 REM      IS THE FIRST SUCH VALUE GREATER THAN THE GIVEN
9800 REM      MAXIMUM (OR THE FIRST VALUE LESS THAN THE
9810 REM      GIVEN MAXIMUM IF THE GIVEN MAXIMUM IS LESS
9820 REM      THAN THE GIVEN MINIMUM).
9830 REM
9840 REM      IN A FEW CASES, THE PROCEDURE WILL RESULT IN
9850 REM      ROUND MINIMUM AND MAXIMUM VALUES WHICH DO NOT
9860 REM      INCLUDE THE GIVEN MAXIMUM VALUE.  IN SUCH
9870 REM      INSTANCES, TRANSFER IS MADE TO LINE 9390 TO
9880 REM      SELECT THE NEXT HIGHEST STANDARD INCREMENT
9890 REM      MANTISSA, AND THE PROCESS IS REPEATED WITH A
9900 REM      NEW ROUND INCREMENT TO GET THE NEW ROUND
9910 REM      MINIMUM AND MAXIMUM VALUES.  IF THE TABLE OF
9920 REM      STANDARD INCREMENT MANTISSAS HAS BEEN
9930 REM      EXHAUSTED, THE CHARACTERISTIC IS INCREASED BY
9940 REM      1, AND THE SCAN OF STANDARD INCREMENTS STARTS
9950 REM      OVER AT THE BEGINNING OF THE TABLE.
9960 REM
9970      LET T2 = A  +  (C * L)
9980      IF  R < 0.0  THEN 10080
9990      IF  T2 >= B2  THEN 10090
10000      IF  ABS((B2 - T2) / R) <= 0.0001  THEN 10090
10010      IF  K < M  THEN 9390
10020 REM
10030 REM      RAISE THE CHARACTERISTIC BY 1 AND GO BACK TO
10040 REM      RECOMPUTE A NEW ROUND INCREMENT.
10050 REM
10060      LET N = N + 1
10070      GO TO 9150
10080      IF  T2 > B2  THEN 10000
10090      LET B = T2
10100      RETURN
10110      END
```

Disk Addition

Add a floppy disk to Motorola's MicroChroma 68 color graphics kit.

Harold A. Mauch
Percom Data Co., Inc.
211 N. Kirby
Garland TX 75042

Motorola developed the MicroChroma 68 kit as an evaluation tool for their 6847 color graphics video display generator, but the MicroChroma 68 is much more than just a color graphics demonstrator. It is a versatile 6800 computing system complete with Kansas City Standard cassette interface, a 2K ROM operating system (TVBUG) and provision for up to 15K of RAM (6K for color display, 8K for user programs and 1K for system use).

Since MicroChroma connects directly to any color television set, all you need to complete the system are an inexpensive keyboard, a cassette recorder and a power supply. For more information about TVBUG and the MicroChroma kit, refer to Tim Ahrens' article in the June 1979 issue of *Kilobaud Microcomputing* ("TVBUG," p. 48).

After you get the MicroChroma kit running, you can add a mini-disk system.

The EXORcisor Bus

The MicroChroma circuit card is designed to plug into and get power from Motorola's 86-pin EXORcisor bus, which is different than the S-100 and SS-50 buses found in hobby computers. However, it is the standard bus for most 6800-based industrial computing equipment. The EXORcisor bus has not been widely accepted by home computerists until recently because the modules cost more than corresponding hobby bus modules.

In addition to Motorola, there are several manufacturers that produce memory and various I/O modules (cards) for the EXORcisor bus. For instance, Percom Data now produces a 64K dynamic memory and a mini-disk system for the EXORcisor.

The Percom Mini-Disk

The Percom EXORcisor bus-compatible mini-disk (LFD-400EX) controller contains all of the ROM and RAM needed to support the disk operating system. This eliminates any burden on the memory resources of the MicroChroma.

Much of the 6800 disk software Percom developed for the EXORcisor and SS-50 bus computers is easily adapted for use with the MicroChroma and TVBUG. The adaptation usually amounts to little more than changing subroutine vectors to match the routines in TVBUG.

Connecting the Parts

Although the MicroChroma card plugs into the EXORcisor bus, the only connection it makes to the bus is for +5 volt power and ground. It is necessary to buffer the address and data bus signals and connect them to the appropriate contacts on the EXORcisor bus connector. The applications booklet supplied with the MicroChroma kit provides all of the necessary information for connecting the kit to the EXORcisor bus. The buffer ICs may be mounted in the wire-wrap area just above the bus connector.

MicroChroma can accommodate more RAM memory than most other evaluation kits; however, you may eventually want to add even more memory. Motorola, Creative Micro Systems and Percom Data manufacture memory cards for the EXORcisor bus.

In the system pictured we added our own memory and disabled the RAM memory on the MicroChroma card by removing the 74LS138 RAM selector chip (U22). Again, the MicroChroma applications booklet provides much useful information for adding and configuring additional memory.

The various circuit cards may be connected together by mounting and interconnecting several 0.156 inch 86-pin connectors. A more convenient connection is possible with the Percom 5-slot motherboard or the Motorola 5- or 10-slot Micro module card cage. We use the Motorola Micro module power supply; however, any of the low-cost open-frame power supplies that deliver 5 volts at 4–6 Amps and ± 12 volts at 1 Amp should be more than adequate.

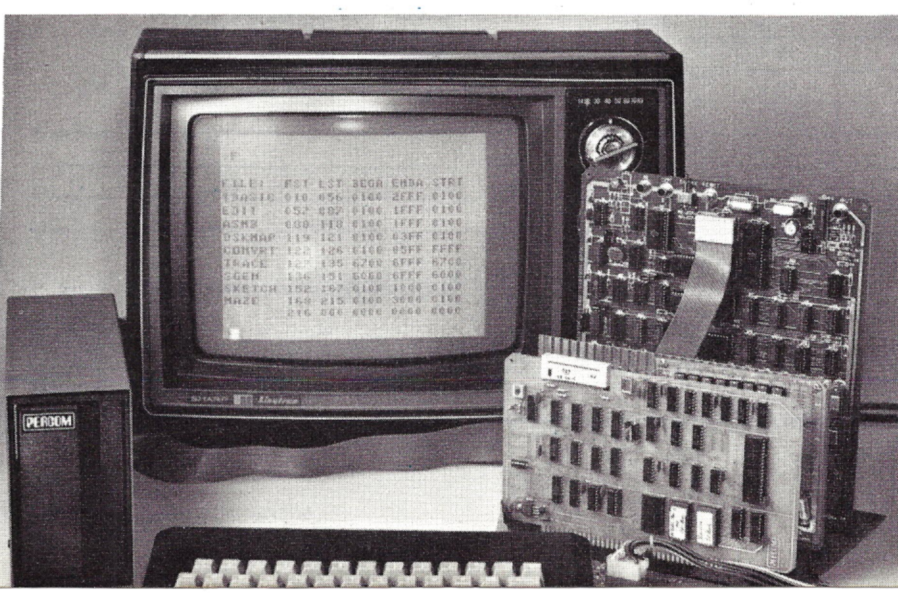
Motorola recommends a modified Cherry keyboard; we use an inexpensive George Risk keyboard. Actually, any ASCII keyboard that generates a low-active strobe that is maintained as long as a key is depressed should work well. Keyboards that generate a short strobe pulse when a key is depressed are unsatisfactory because the keyboard scanning routine in TVBUG may miss the strobe pulse.

Where to Get the Parts

You can purchase the MicroChroma kit from nearly any Motorola distributor; however, the kit only contains the LSI devices and the PC card. Austin Electronics also sells the MicroChroma kit. In addition, they have a kit of all the extra parts needed to complete the kit.

The Percom products are available directly from Percom Data Co., Inc. When ordering, be sure to mention you intend to use the mini-disk with TVBUG.

Creative Micro Systems (CMS) manufactures a broad line of EXORcisor bus-compatible memory cards and I/O modules. The entire CMS line is available from Advanced Computer Products (PO Box 17329, Irvine CA 92713). ■



Percom's system for the MicroChroma 68.

Robert M. Morgan
3449 Brunswick Rd.
Lexington KY 40503

An H8 in the Darkroom

There may be some techniques for using the H8 panel monitor, via BASIC, that you haven't even contemplated. If such is the case, you're going to read about them in this article.

As part of preparing "Heath's H8 Is a Winner!" (*Kilobaud* No. 20, p. 70), I had to submit some black and white photos. While assembling my darkroom equipment, I once again experienced an old yearning shared by all photo printers—a darkroom clock to count on . . . and see . . . and hear while exposing, developing, stopping and fixing.

While I was in this wishful mood, my eyes fell upon my H8's keypad and LED array, and I knew my timer wasn't far away. After a few minutes of BASIC work, the timer was done, negatives printed and package on its way.

Keypad Input

As a prologue to describing the timer program, a few comments are in order about BASIC's ability to communicate with front-panel facilities. Heath's Extended BASIC is normally configured for operator input from the console (Heath H9, ADM 3A, etc.). However, an alternate input facility is BASIC's PAD function. If Extended BASIC were loaded, and you were to enter *PRINT PAD(0), BASIC would lock up until you pressed one of the keys on the keypad—then the value entered would be displayed on the console, and would be followed by the * prompt character. See Fig. 1 for an example of this.

Try the routine in Fig. 2 to discover what number is assigned to each key—remember that should you enter a CONTROL-C to break out of the routine, BASIC will still wait for the key entry on line 10 *prior* to servicing the CONTROL-C (as with most Heath software products, BASIC accepts *all* console input as interrupts, and in this case the CONTROL-C is held until the PAD function is completed by pressing any key on the keypad). The assignments are shown in Fig. 3.

BASIC uses PAM8's (Heath Panel Monitor) routine for reading the keypad, so each entry is debounced and followed with a "click" in the front-panel speaker.

LED Output

Heath has thoughtfully provided a BASIC function called SEG to return a numeric value appropriate to displaying a zero-through-nine digit on one of the 7-segment LEDs. Function SEG will not accept negative arguments, and arguments greater than nine will generally result in the digit 8 being displayed. SEG will accept a floating-point argument and will truncate it down to the next integer.

Try using SEG with the program in Fig. 4. Line 10 turns on a bit in PAM8, which allows a user program to write into the LEDs with the POKE function. After a value is input by line 20 and its SEG value displayed, the SEG value is POKEd in location 040.013 (decimal 8203), which is the high-order address LED (left-most LED). Either the command-form or statement-form CNTRL 2,0 will once again blank the LEDs; this is their normal mode when BASIC is executing.

To view all possible LED segment combinations, enter and execute the program in Fig. 5. Lines 20-40 blank out the LEDs, and a FOR loop is entered. After the current value of loop index "I" is displayed in the left-most LED, the index is decoded into hundred-ten-unit values that are displayed on the three right-most LEDs.

Because of line 130 the loop will pause until the space bar on the console is pressed (with-

Table 1. LED segments displayed from several decimal values.

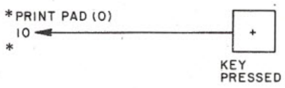


Fig. 1. Simple keypad test.

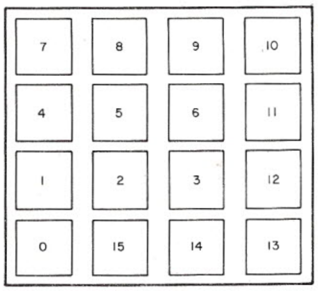


Fig. 3. Key assignments.

```
10 PRINT PAD(0)
20 GOTO 10
```

Fig. 2. Pressing keys will confirm values.

```
10 CNTRL 2,1
20 INPUT I
30 PRINT SEG(I)
40 POKE 8203,SEG(I)
50 GOTO 20
```

Fig. 4. Test of SEG function.

```
10 CNTRL 2,1
20 FOR I=0 TO 8
30 POKE 8203+I,255
40 NEXT I
50 FOR I=0 TO 255
60 POKE 8203,I
70 H=INT(I/100)
80 POKE 8209,SEG(H)
90 T=INT((I-H*100)/10)
100 POKE 8210,SEG(T)
110 U=INT(I-H*100-T*10)
120 POKE 8211,SEG(U)
130 PAUSE
140 NEXT I
```

Fig. 5. LED combination program.

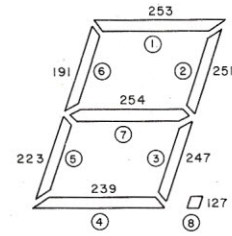


Fig. 6. LED segments and values.

out the pause, each combination is displayed less than 0.1 second). As you continue to examine these patterns, you'll find many to be nonsense and a few that may be useful in future applications. Table 1 illustrates some that may be of interest.

Fig. 6 identifies the decimal value required to turn on each segment in an LED if POKEd to the correct address in the CNTRL 2,1 mode. These values can be placed in an array (S(1) = 253, S(2) = 251, S(3) = 247 . . .) and combined by the AND function to form any desired segment combination. For example, the letter "J" could be displayed by J = S(2)AND S(3)AND S(4)AND S(5) followed by POKE 8203,J.

Speaker

Heath's Panel Monitor (PAM8), everpresent in the first 1K of ROM and a few bytes of RAM, includes in its treasure of goodies a subroutine called HORN at 002.140. When executed, it will turn on the 1000 Hz panel speaker for x milliseconds, where x is the current value in register (A) times 2. To use in BASIC, configure Extended BASIC with some space at the top of your RAM for a USR routine, load the routine to be memory resident with BASIC, and set Extended BASIC's USR pointer to the start of your routine.

Assuming you have 16K of RAM and you've left 1K at the top for USR routines when you did your BASIC configuration (SYSGEN), the six-byte USR function in Fig. 7a will allow BASIC access to the speaker. Since these six bytes can be easily represented in decimal, as shown in Fig. 7b, the USR function can be loaded by a simple POKE to the correct address (134.000 = decimal 23552). The POKE is also suitable for configuring BASIC to point to 134.000 as the USR function (do this with POKE 17268,92).

A couple of observations on USR functions: Although Heath doesn't step you through their use in the H8 *Software Reference Manual*, there is a wealth of hints that can be dug out regarding use of internal BASIC routines. Also, since BASIC can be dynamically reconfigured to change the pointer to the USR function through POKEs to 17267 and 17268, several USR functions can be in high memory and the appropriate one selected by correct loading of the USR pointer.

The Timer

The Program listing combines these three features of the H8 (keypad input, LED output and speaker) with PAM's

Program listing.

```
10 REM PHOTO TIMER USING H-8 LED'S, KEYPAD, AND FRONT-PANEL SPEAKER.
20 REM
30 REM FOLLOWING LOADS A USR ROUTINE TO CLICK FRONT-PANEL SPEAKER
40 POKE 17268,92
50 DATA 62,1,205,96,2,201
60 P9=23552
70 FOR I=0 TO 5
80 READ I1
90 POKE P9+I,I1
100 NEXT I
120 REM P9 SET TO POINT TO MEMORY LOCATION THAT SETS 'CLICK' LENGTH.
130 P9=P9+1
140 DIM M(3),Q(3,9)
150 FOR I=0 TO 9
160 Q(1,I)=SEG(I)
170 Q(3,I)=SEG(I)
180 NEXT I
190 DATA 1,115,72,96,50,36,4,113,0,32
200 FOR I=0 TO 9:READ Q(2,I):NEXT I
210 CNTRL 2,1
220 REM "A" POINTS TO FIRST LED, AND "C" TO HIGH-ORDER BYTE OF CLOCK.
230 A=8203
240 C=8220
250 REM BLANK-OUT DIGITS
260 GOSUB 5000
1000 REM ====================================================
1010 REM            C O M M A N D    L O O P
1020 REM ====================================================
1030 REM   "+" COMMAND (VALUE 10) = EXECUTE COUNT
1040 REM   "/" COMMAND (VALUE 13) = EXECUTE UPDATE
1050 X1=PAD(0)
1060 IF X1=10 THEN GOSUB 3000
1070 IF X1=13 THEN GOSUB 2000
1080 IF (X1=10 OR X1= 13) THEN 1050
1090 REM == ERROR - SOUND BELL AND TRY AGAIN
1100 PRINT CHR$(7);
1110 GOTO 1050
```

```
2000 REM ====================================================
2010 REM SUBROUTINE TO UPDATE COUNTER
2020 REM 3 DIGITS REQUIRED.....LAST MUST BE '5' OR '0'
2030 REM BLANK DISPLAY
2040 GOSUB 5000
2050 REM GET THREE NUMBERS FROM KEYPAD AND STORE IN ARRAY "M"
2060 REM           IN ORDER  3,2,1
2070 X=PAD(0)
2080 IF X<10 THEN 2130
2090 REM ERROR: VALUE MUST BE 0 TO 9
2100 PRINT CHR$(7);
2120 GOTO 2070
2130 Q=Q(3,X)
2140 M(3)=X:POKE A,Q
2160 X=PAD(0)
2180 IF X<10 THEN 2230
2200 PRINT CHR$(7);
2220 GOTO 2160
2230 Q=Q(2,X)
2240 M(2)=X:POKE A+1,Q
2260 X=PAD(0)
2280 IF (X=0 OR X=5) THEN 2330
2300 PRINT CHR$(7);
2320 GOTO 2260
2330 Q=Q(1,X)
2340 M(1)=X:POKE A+2,Q
2360 Z1=M(1):Z2=M(2):Z3=M(3)
2999 RETURN
3000 REM ====================================================
3010 REM COUNTS DOWN DIGITS IN ARRAY "M"
3100 SET COUNTER TO ZERO
3120 POKE C,0: POKE C-1,0
3140 Z=1: GOTO 3180
3145 REM TIMER LOOP:
3150 GOSUB 6000
3180 X=PEEK(C)
3200 IF X<>Z THEN 3180
3205 U2=USR(U2)
3210 Z=Z+1: IF Z>255 THEN Z=Z-256
3215 M(1)=M(1)-5
3220 IF M(1)>=0 THEN 3150
3225 M(1)=5
3280 M(2)=M(2)-1
3300 IF M(2)>=0 THEN 3150
3320 M(2)=9
3340 M(3)=M(3)-1
3360 IF M(3)>=0 THEN 3150
3380 POKE P9,20:U2=USR(U2)
3390 M(1)=Z1:M(2)=Z2:M(3)=Z3
3410 GOSUB 6000
3420 POKE P9,1
3999 RETURN
5000 REM ====================================================
5010 REM BLANK OUT LED'S
5020 FOR I=A TO A+8
5030 POKE I,255
5040 NEXT I
5999 RETURN
6000 ========================================================
6010 REM UPDATE DIGITS
6020 I=A
6030 FOR Q=3 TO 1 STEP -1
6040 POKE I,Q(Q,M(Q))
6050 I=I+1
6060 NEXT Q
6999 RETURN
9999 END
```

interrupt-driven clock to provide a timer that is easy to use in the dark, and quite repeatable.

Repeatability rather than absolute accuracy is stressed; for both black and white and color, most darkroom workers establish procedures based on conditions that they know can be obtained for each session. For example, I *always* make my contact sheets with the film plane of the enlarger 24 inches above the easel, lens set for f3.5 and in focus, negative carrier empty, RC paper, 3½ seconds exposure and developer at about 75° F—this works *every* time.

My approach on the timer program was to use the high-order byte of the 16-bit clock. Its least significant bit is incremented by an interrupt every 2 ms, so the least significant bit in the high-order byte (the ninth bit in the 16-bit register) is incremented every 0.512 seconds. This provides crystal-controlled repeatability at 2.4 percent absolute accuracy, so a 60-second countdown will really take 61.44 seconds... but it will do it every time!

BASIC just doesn't process fast enough to use the low-order byte. To prove this for yourself, execute the following series of repetitive PEEKs at the low-order byte: FOR I = 1 to 1000: PRINT PEEK(8219);:NEXT I. You'll get numbers like 74 89 104 122 140 158 175 194 212 231 249 11 27 and so on, if you're running your console at 4800 baud—the spread will be greater if your console is slower.

Now substitute 8220 in the PEEK and reexecute; at 600 baud you'll see six to ten duplications of the high-order byte before it is incremented (at 9600 baud, 16 to 23 duplications). It should be clear that the low-order byte cannot be processed with BASIC.

In lines 10-100 of the timer program the USR routine is loaded and BASIC configured for USR with the POKE to 17268 (134.164). P9 is set to modify the second byte in the MVI instruction (via a POKE), which loads register (A) with the time the horn is to be turned on.

In lines 140-200 a matrix Q is defined. Its first and third rows are used to store the appropriate values to display integers on the LEDs. Since the timer's range is 99.5 to 00.0, the second row of Q includes data from line 190 to display integers *with* decimal points. After the LED write bit is turned on, and pointers to the LED's RAM and the clock are established, we enter the command loop.

The program is set up so that as soon as you enter RUN, you can turn off the CRT console. All program functions are controlled by the keypad. Line 1050 reads a keypad value, which is subsequently tested. Should the value be in error, the console BELL is sounded, if the console is on, and the program waits for a correct entry—either upper-right or lower-right keys.

The UPDATE function should be first selected to place a value in the counter array "M." Upon entry into UPDATE at line 2000, the display is blanked via subroutine 5000, and three key entries are processed. Should they be in error (i.e., not be values 0-9 for the first two, or 0 or 5 for last), the BELL will sound (if the console is on) and that digit will not display on the LED panel; otherwise the selected digit will display. After the third correct key entry, three scalers that save the digits will be updated (Z1, Z2, Z3) and control returned to the command loop.

Upon entry to COUNTER, the clock bytes are zeroed to maximize accuracy of the first half second, and the target ("Z") for comparison to the high-order byte set to 1. The high-order clock byte is then tested until it equals the target, which is a delay of 512 ms. When they are equal the speaker is "clicked" (useful if the printer is "dodging" or "burning" the print) and the target incremented.

The low- to high-order values in the counter array are decremented, as required, and the display updated via subroutine 6000. The timer loop is reentered at line 3180. When the "tens" member of the counter array goes negative, time is up and the subroutine is exited after sounding a 10 ms beep on the speaker.

What's Missing

Direct control of the enlarger and safelight would be an obvious enhancement. Announcement by Mullen Computer Boards of an H8 Relay/Opto-Isolator Control Board should make this possible. Although I have no experience with the Mullen board, I believe the program modifications shown in Fig. 8 should provide the required control. These modifications assume that a 2 to port 144_8 would activate relays to turn on the safelight and off the enlarger while a 1 to port 144_8 would turn off the safelight and on the enlarger.

Another enhancement: To stop the count while it is counting, add line 3190—IF PIN(240) <> 255 THEN 3380. Pressing any key on the keypad will cancel the count.

Wrap-up

You've reviewed how some H8 PAM facilities are available to the BASIC programmer. Whether you are counting, clocking, displaying error or record numbers, or alpha messages such as 140,222,222,198, 222,254,254,254,254 (Error----), or warning a data-entry operator with a speaker click, beep, or howl—these facilities are available for the creative designer. ■

```
134.000  076 001        MVI   A,001Q
134.002  315 140 002    CALL  HORN
134.005  311            RET
```

Fig. 7a. USR program to interface with speaker.

```
Octal:     076, 001, 315, 140, 002, 311
Decimal:    62,   1, 205,  96,    2  201
```

Fig. 7b. Decimal values for POKEing.

```
 270 OUT 100,2
1035 REM  "*" COMMAND (VALUE 10) = EXECUTE FOCUS
1065 IF X1=12 THEN GOSUB 4000
1080 IF (X1=10 OR X1=12 OR X1=13) THEN 1050
3130 OUT 100,1
3370 OUT 100,2
4000 REM ====================================
4010 REM FOCUS SUBROUTINE
4020 OUT 100,1
4030 REM WAIT FOR ANY KEYPAD ENTRY TO GO BACK.
4040 X=PAD(0)
4050 OUT 100,2
4999 RETURN
```

Fig. 8. Probable mods to add to Mullen Relay Board.

TBS-80
GENERAL ACCOUNTING SYSTEMS.
ONE STEP BEYOND.

Your TRS-80™ microcomputer is not a toy. These **TBS-80 general accounting systems** aptly demonstrate the power of your computer.

ANALYSIS PAD *by Del Jones is the epitome of first-class programming in business applications. Requiring 48K, and one disk with a printer recommended, this columnar calculator gives the user tremendous flexibility in data entry, enabling the user to create 30 or more columns and rows. Enter your own column and row labels. Enter your data by row or column or directly onto screen display via edit mode. Move, swap, delete, and add rows or columns. Create new pads by stripping relevant data from old files. You never have to key in data twice. But, more important than the powerful data manipulation provided, add, subtract, multiply and divide one column by another and put results in another column. Perform up to six calculations on one column and even define one column to be a constant. The calculation routine you create can be saved and reused. Print out the entire pad in four column segments to line or serial printer.* ***ANALYSIS PAD*** *was originally advertised for 32K tape at $32.50. Since then, it has been totally rewritten and expanded to its present 48K disk only form and sells for $49.50. It is easily worth twice as much. You have to see it to believe it.*

CHECK REGISTER ACCOUNTING SYSTEM, *adapted for the TRS-80 by Dale Kubler and originally written by O.E. Dial, is the most comprehensive check-balancing program written. Requiring 32K, two disks and printer, this program does much more than just balance and reconcile your checkbook. It enables you to define up to 60 account names and will generate monthly summaries of all accounts with monthly and year-to-date totals. Single-entry input allows the user to disperse one transaction over several accounts and to make a 64-character note on each transaction. Checks can be printed out after data has been entered. Aside from the Statement of Accounts,* ***CRAS*** *also generates the following reports: Check Register for any Month, Notes to Check Register, Income/Expense Distribution, Statement of Selected Accounts, Bank Reconcile Statement and Suspense File. The Suspense file is an extra feature where you can make notes to yourself for any month in the year.* ***CRAS*** *will make both you and your account happy and it sells for $49.50.*

CHECKBOOK II *by Alan Meyers is the finest program of its kind yet published. With superb graphic screen displays, it does everything necessary to keep your checkbook balanced. Data is input directly into a five-column screen display with a field for alpha or numeric codes. Editing is done easily for changes in any or all columns.* ***CHECKBOOK II*** *will accurately balance and reconcile your checkbook, handling balances up to $1,000,000. Your balance brought forward is always in memory. Outstanding checks are listed and easily saved. You can also search for an entry by any field except amount, and all checks with matching entries will be displayed and totaled. A numeric sort routine is included. Screen prints can be made to a line printer from almost any point in the program. In addition, the 32-48K version can write files to disk. This, and the 16K version, are included on the same tape. For $18.50,* ***CHECKBOOK II*** *is the bottom line in personal checkbook programs. A disk version of this program is available for $28.50.*

BUDGET II *(not yet released) by Alan Meyers, takes off where CHECKBOOK II ends. Written exclusively for either disk or tape based computers, this program enables the user to set up 20 account names with four character codes for each, that correspond to the codes used in Checkbook II. Each account can be tagged income or expense and whether it is fixed or not. Set your monthly budget and balance it. Disperse your cash account over the other accounts. Checkbook II data is brought in and summarized by account and compared to amount budgeted. Year-to-date totals are included in monthly summary. Year Summary gives monthly and year totals for each account at a glance. Forecast feature enables user to enter rate of inflation and income increase to see financial standing after 12 months. Review enables user to go back and look at months previously summarized. Flashy graphics and much more. For 16K and 32K tape,* ***BUDGET II*** *sells for $24.50. For 32K up disk, $34.50. If you have CHECKBOOK II, you will want this program.*

TBS *has other incredible software for Tandy's microcomputer. Intent on making it a powerful tool, we have* ***large scale business accounting systems, data processing systems, system utilities,*** *and the* ***Library 100.*** *We have the only* ***DISK HEAD CLEANER*** *(for APPLE too!) and* ***GRAN MASTER DISKETTES,*** *the best on the market.*

TBS *is* ***YOUR COMPANY,*** *and we build systems, not just software. The above products are available now, nationwide. Visit your local Computer Dealer or Associate Radio Shack Store and demand the best, demand TBS. For more information, contact us through the numbers below.*

™TRS-80 is a trademark of the Tandy Corporation.

TBS™
THE BOTTOM SHELF, INC.
(404)939-6031 • P.O. Box 49014 • Atlanta, GA 30359

33

Reader Service—see page 258

Expand Your H8

Expand your Heath H8, use it in the darkroom . . . or both.

Larry Spani
William Seely
Syntext Data
PO Box 24
Monmouth OR 97361

If your H8 motherboard is full and there are additional boards you have your eye on, then read on.

We purchased our H8 system in December 1977 and quickly filled up all of the available slots, including the CPU board, disk controller board, two serial boards and four 8K memory boards. The 16K board was not yet available (we weren't even aware that they had one planned), but by the time it was available, we were sorry we did not have the foresight to see it coming.

This brought us to the problem at hand: only 32K of memory and no room for expansion. One alternative would have been to replace three of the 8K boards with 16K boards. This would have been both costly and left us with three less-than-useless 8K boards. Expanding the motherboard seemed a more logical approach.

We checked with the Heath computer technicians and found there were no plans in the works for expanding the H8 and probably never would be. One of the popular board manufacturers had also mentioned the possibility of an expansion board for the H8, but after writing to them we found that they, too, had no immediate plans.

If we wanted this expansion, we would have to do it ourselves; so we did. For approximately $180, we designed our own expansion interface, and we are pleased with the results. If you are in a similar situation, or if you just want the option of having the room for future board releases, then this article is for you.

Description

The expansion interface consists of a printed circuit board, expansion cabinet and an interconnecting cable (Photo 1). The expansion unit was designed primarily to accommodate memory boards (8K and/or 16K) only. When you are finished, you can move memory from the H8 to the expansion unit with ample room in the H8 for all other boards. Because of this, we found that it was not necessary to design the expansion PC board to accommodate all Heath boards. Due to their variety, this would have been much more involved.

We strongly suggest that you

CABINET
Refer to your H8 assembly manual. Order the items shown in the Parts List under Chassis Assembly with the *exception* of the following:

ELECTRONIC COMPONENTS	Speaker
SHEET METAL PARTS	Circuit board mounting bracket (2)
	Circuit board support clamp
PRINTED MATERIAL	Do not order any of these items
MISCELLANEOUS	PCB connectors (2)
	Nut Starter
	IC Lifter
	Solder Wick
	Window
	3-Ring Binder
	Set of 5 Tabs
	System Software
PC BOARD	
Parts Needed	(1) 6" × 6" double sided copper board
	(2) 25-pin connector (Heath P/N 432-947)
	(2) 25-pin 90 degree header strips
	(3) 20-pin IC sockets
	(3) 74LS241
	(1) +5 Volt regulator (7805)
	(1) 2.2 uF tantalum capacitor (20 V)
	(1) .01 uF capacitor (100 V)
	(3) 1.2k Ohm ¼ Watt resistors (10%)
MISCELLANEOUS	
Parts Needed	2½' # 14 wire pair
	2-pin Molex Connector
	18" 50-conductor ribbon cable
	2 25-hole connector shells (Heath P/N 432-948)
	50 spring clips (Heath P/N 432-866)

Table 1. Parts list.

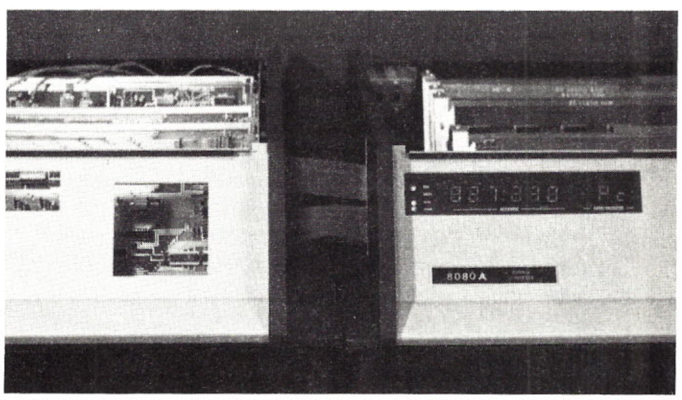


Photo 1. The interconnecting cable tying the two units into one.

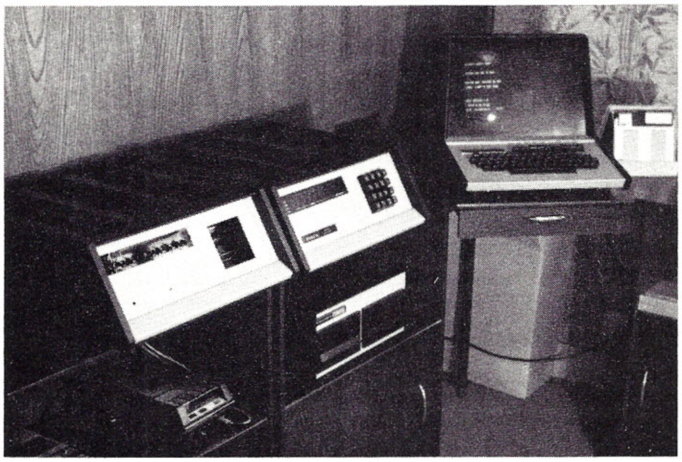


Photo 2. H8 system with expansion unit.

A0	through A3	IC2,(11-9),(13-7),(15-5),(17-3)
A4	through A7	IC1,(11-9),(13-7),(15-5),(17-3)
A8	through A11	IC1,(2-18),(4-16),(6-14),(8-12)
A12	through A15	IC2,(2-18),)(4-16),(6-14),(8-12)

Table 2.

(17-3)	RESET	(15-5)	MEMORY READ
(13-7)	I/O READ	(11-9)	MEMORY WRITE
(2-18)	PHASE Ø2	(4-16)	I/O WRITE
(8-12)	M1	(6-14)	NOT USED

Table 3.

read through the entire article and assess your situation before proceeding with any one part of this project.

The cabinet consists almost totally of Heath parts, for several reasons. The most prominent is cost. After pricing an appropriate enclosure and power supply, we found that those two items alone were equal to the cost of a complete H8 in parts, minus the CPU, front panel boards and a few other miscellaneous parts. There were also the problems of design (critical) and appearance. By sticking with the H8, we were able to solve all of those problems with a minimum of effort and cash.

A complete list referencing all needed parts is shown in Table 1. You may already have some of the items; you might purchase the others locally at a reduced cost, but be careful not to compromise quality for a few cents. If you do not have a price list of the parts for the H8, write or call Heath for one.

The double-sided printed circuit board is 6×6 inches. This board costs approximately $30 to make, but it may be less for you if you already have developing and etching materials.

The cabinet will take most of the time needed to complete this project, as there are modifications to make in addition to assembling the cabinet.

Circuit Description

The circuit is a concise buffering unit composed of three noninverting octal buffer/line drive ICs (74LS241). Two of the ICs are used to buffer the 16 address lines, and the remaining IC buffers the memory write, I/O write, memory read, I/O read, phase Ø2 clock, reset and M1 (see Fig. 1).

Keep in mind that the 50-pin bus begins its count with 0, not 1. References to ICs will be given with the pin IN number first, followed by the corresponding pin OUT number, using the format IC3 (4-16).

IC1 and IC2 buffer the address lines as the signals move across the board. The address lines are buffered as shown in Table 2. If you use wire wrap rather than a printed circuit board, A0–A7 may be accommodated by one IC, and A8–A15 by the other. This will reduce the amount of wire crossover.

IC3 buffers the remaining signals as shown in Table 3. All other lines are unbuffered and are either carried by the foil pattern or by jumper.

The remaining pins are common on all three ICs and are assigned as shown in Table 4. The H8 and the expansion unit each have three voltage supplies. While the +8 volt supplies are connected by a #14 wire pair (described later), the remaining two supplies (+18 and −18) are interconnected between the two units via the foil pattern on the PC board.

The PC Board

The PC board (Photo 2) is relatively easy to make. The patterns for each side of the board are shown in Fig. 2. There are numerous books and articles to aid you in making PC boards.

Once the PC board is etched and drilled, test all of the patterns on the board for continuity. It is much easier to correct a foil pattern problem now than later. Once you are sure the board is sound, the parts can be added.

Solder the three IC sockets into place. Next, insert and solder in the two Heath connector shells (P/N 432-947) at S1. Then insert the two 25-pin 90-degree header strips into S2 and solder those into place. Connect jumpers in the locations shown in Table 5.

Solder in a .01 uF capacitor at C2 and a 2.2 uF tantalum capacitor at C1. Install 1.2k Ohm resistors at R1, R2 and R3. Install the 7805, +5 volt regulator into its location at the high end of S1. Jumper remaining holes in the board to the other side by inserting small lengths of wire through the board and soldering on both sides. Cut off all excess lead lengths. Recheck your work for poor solder connections, solder bridges, etc. A continuity check ensures proper installation of all parts. Insert the three 74LS241s into their sockets, and be sure pin 1 is in the proper place. This completes the PC board.

The Enclosure

Once you have all of the parts, put the expansion cabinet together using your H8 assembly manual for guidance. Three modifications must be made. Two involve wiring changes, and the third is a modification in the cabinet.

Since the H8 and the expan-

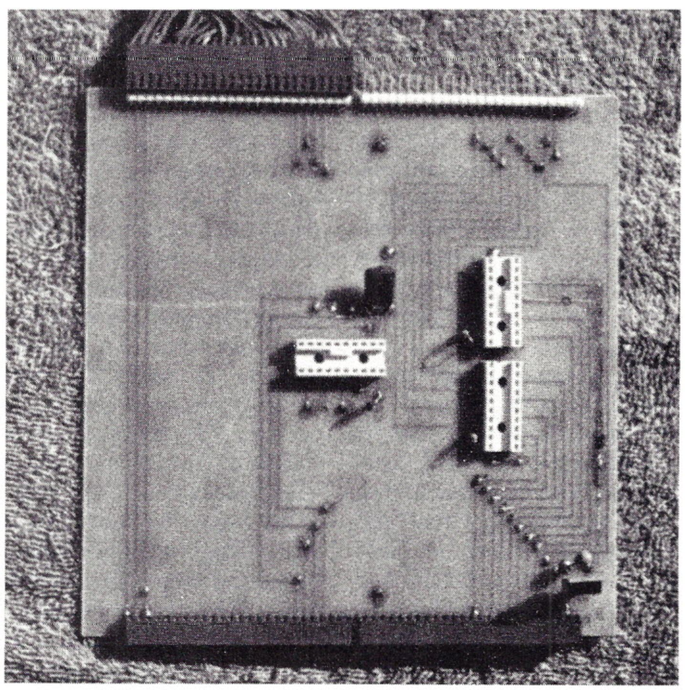


Photo 3. PC board for expansion unit.

Pin 10	is grounded and is tied to ground.
Pin 20	is connected to the +5 volt supply.
Pin 1	is an inverting control gate used to control the operation of four of the internal buffering circuits in each IC. This pin is tied to ground.
Pin 19	is a non-inverting control gate used to activate the remaining four internal buffering circuits in each IC. This pin is tied to the +5 volts at Pin 20 through a 1.2k Ohm resistor.

Table 4.

(A) IC3 PIN 18	to	(A) S2 PIN 22
(W) IC3 PIN 16	to	(W) S2 PIN 21
(M) IC3 PIN 12	to	(M) S2 PIN 19
(R) S1 PIN 20	to	(R) S2 PIN 20
(D) S1 PIN 27	to	(D) S2 PIN 27
(P) S1 PIN 47	to	(P) S2 PIN 47

Table 5.

sion unit must have power applied at the same time, it will be necessary to change the ac power switch wiring as follows (these steps follow the assembly instructions in your H8 manual; also refer to Fig. 3):

Do not connect a wire between lug 1 of the power switch and lug 2 of the fuseholder.

Do not connect the black line cord lead to lug 1 of the ac socket; instead, connect it directly to lug 1 of the power switch.

Do connect a wire between lug 2 of the power switch and lug 1 of the fuseholder.

Later in the assembly manual you are instructed to connect a lead from one of the switches located in the ac shield to lug 2 of the power switch; instead, connect this lead directly to lug 2 of the fuseholder.

When the project is complete, plug the H8 directly into the ac socket of the expansion unit, and the H8 power switch must be in the on position permanently. When you turn the expansion unit on, both units will have power applied at precisely the same moment.

The other wiring modification involves tying the two units' +8 volt power supplies together for consistency in operation. To do this, use a #14 wire pair to connect the terminals of the 77,000 uF capacitors of each unit (+ to +, − to −). Two and one-half feet of wire will be sufficient to reach between the two capacitors using the existing cable openings in the rear of each unit. We also put a Molex connector in the middle of the wire pair for ease of disconnection of the two units if it ever becomes necessary.

The third modification involves cutting two slotted openings—one in the right side panel of the expansion unit and the other in the left side panel of the H8. The interconnecting cable will pass through these openings. Each slot is directly in line with P3 of the motherboard and has the same angle. It will be necessary to remove all boards in the H8 to do this, and the motherboard in the expansion unit must not be in place.

Drill two holes in each of the two side panels. The top hole is measured 1 inch down from the top and 4½ inches in from the front edge of the side panel (refer to Fig. 4). The second hole is both 4½ inches down from the top and 4½ inches in from the front edge. Use a small bit to drill pilot holes, then redrill the holes with a ¼ inch bit. Once this is done, use a straight edge to draw two lines between the outer edges of the holes. Use a coping saw with a fine-toothed saw blade to cut out the area between the holes. The side panels are made of plastic and cut easily. If necessary, use a fine file to even any rough spots when you are finished cutting.

The reason for cutting these slots is that the interconnecting cable must not be over 18 inches long, or buffering problems will occur. If you do not wish to cut into the side panels, you can go over the top of them, but this still means you have to niche

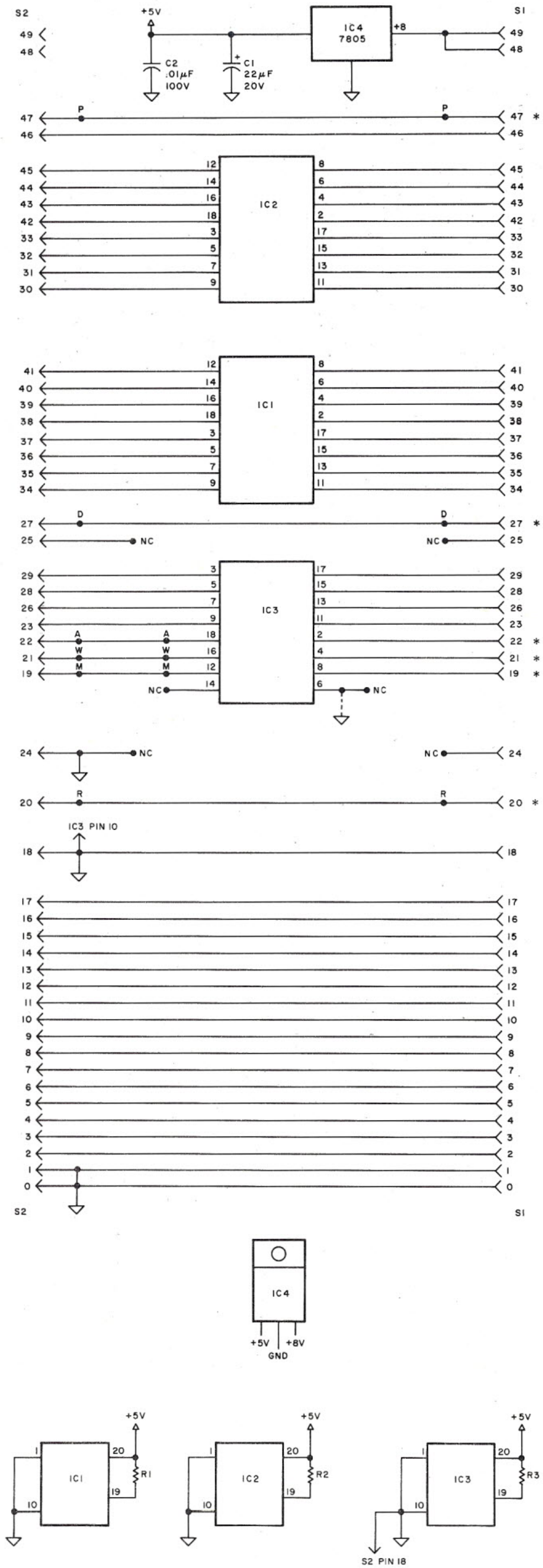
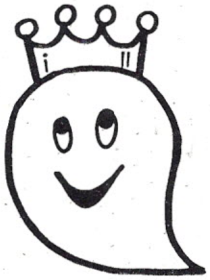


Fig. 1. Schematic for PC board.

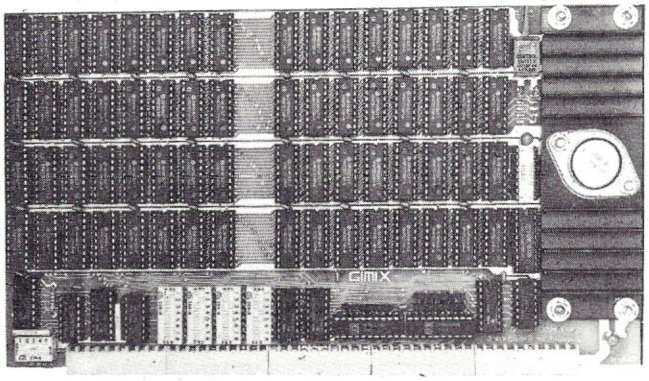


WANT TO BE KING OF THE HILL?

TREAT YOURSELF ROYALLY WITH GIMIX UNIQUE AND INCOMPARABLE BOARDS AND SYSTEMS... DIP-switch Versatility for use with both SS50 (6800) and SS50C (6809) Systems (SWTP. etc.)

32K STATIC RAM BOARDS

Designed for use with: ★ **Existing SS50 Systems**
★ **SS50C Extended Address Systems**



- Assembled
- Burned In
- Tested

16K $328.12

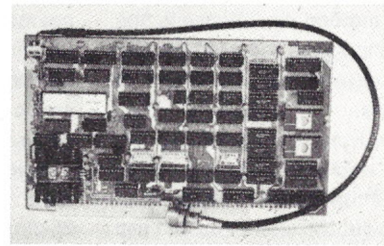
24K $438.14

32K $548.15

16K and 24K Versions are socketed for 32K and require only additional 2114's for expansion.

FEATURES:

- Decoding for 4 Extended Address Lines (allows memory decoding up to 1 megabyte)
- DIP-switch to set extended addressing or disable it
- 4 separate 8K blocks, addressable to any 8K boundary by DIP-switch
- Each 8K block may be individually disabled
- Write protect either of two 16K sections
- Low power consumption — uses 2114L low power RAMS — (2 amps typical for 32K)
- Fully Socketed
- Gold Bus Connectors
- Guaranteed 2MHz operation



THE UNIQUE GIMIX 80 x 24 VIDEO BOARD

★ Upper and Lower Case with Descenders
★ Contiguous 8x10 Character Cells
★ Hardware Scrolling
★ X-Y Addressable Hardware Cursor

It is the ***ONLY* Video Board** that gives you **Software Control** of:

- A programmable RAM Character Generator plus 2 EPROM Character Generators (128 char. ea.).
- Selecting 256 Displayable Characters from 384 available.
- Normal or inverse video, full or reduced intensity, or combinations of these by ***both*** ASCII Code and Bit 8.
- GHOSTability — multiple boards at the same address.

Fully decoded, occupies only 2K of address space
Fully socketed — Gold bus connectors.
Assembled, Burned-in, and Tested at 2MHz.

Deluxe Version **$458.76**
Without RAM Character Generator . **$398.74**
Other Video Boards from **$198.71**

FACTORY PRIME STATIC RAMS		
	2114L 450 ns **$5.90**	200 ns $6.90
	4044 450 ns **$5.90**	250 ns $6.90

ADD $5.00 HANDLING ON ORDERS UNDER $200.00

I.O. BOARDS

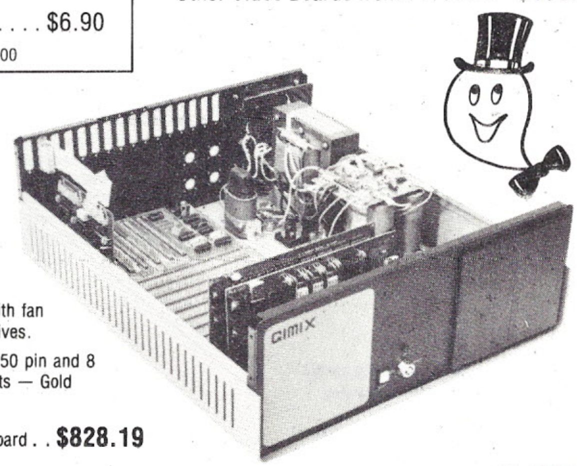
NEW!! 50 PIN 8 PORT RS-232 SERIAL BOARD

★ With SS50C extended addressing (can be disabled)
★ DIP-switch selectable Baud Rates for each Port
★ Occupies 16 bytes of address space
★ Dip switch addressable to any 16 byte boundary
★ Any 2 of IRQ, NMI or FIRQ can be used
★ RS-232 handshaking for each port (CTS, RTS, DCD) **$288.40**
★ With optional on Board Baud Rate Generator **$318.46**

ALSO AVAILABLE

50 Pin 8 Port Parallel **$198.45**
30 Pin 2 Port Parallel . . . . **$ 88.42**
30 Pin 1 Port Serial (RS 232 or 20MA, current loop) . . . . **$ 88.41**
Cable Sets for above . . . . ea. **$ 18.95**

THE CLASSY CHASSIS



- Ferro-resonant Power Supply
- Heavyweight Aluminum Cabinet with fan and provisions for two 5'' disk drives.
- 6800/6809 Mother Board, fifteen 50 pin and 8 DIP-switch addressable 30 pin slots — Gold Plated Pins. Fully decoded.
- Baud Rate Generator on Mother Board . . **$828.19**

32K SYSTEM *Incomparable Features at a Comparable Price!* **$1,594.59**
Includes: Chassis, 6800 CPU, 32K RAM Board, Choice of I/O Card.

16K Version of above **$1,374.49**

EXPORT NOTES:

For 50Hz 230V C.V. POWER SUPPLY . . . . Add $30.00
80 x 24 VIDEO BOARDS — Specify Format (No Added Charge)
On Orders under $250.00 for a Single Board, or Chips, please Add $30.00 Handling and we will ship Air Mail Prepaid.
On all other orders we will ship via Emery Air Frieght Collect, and we will charge no handling. All orders must be prepaid in U.S. Funds. Please note that foreign checks have been taking about eight weeks for collection, so we would advise wiring money or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, account #73-32033. Visa or Master Charge also accepted.

Phone, write, or see your dealer for details and prices on our broad range of Boards and Systems for the SS50/SS50C bus and our AC Power Control Products for all computers.

GIMIX Inc.

✓22

The Company that delivers
Quality Electronic products since 1975.

1137 WEST 37th PLACE, CHICAGO, IL 60609
(312) 927-5510 • TWX 910-221-4055

out the tops of the panels to allow the covers to fit. Unfortunately, the cable is not long enough to use the existing cable openings in the rear of each unit. We found the slotted-openings technique best, but whatever method you choose is fine, as long as the cable length does not exceed the maximum 18 inches.

Next, cut a 50-conductor ribbon cable to the appropriate length; 14 inches is plenty if you have cut the slots. You can separate the cable into two 25-wire cables for ease of handling. On one end, solder a spring clip to each of the 50 wires. Then fasten the wires into two 25-pin female connector shells. On the other end, separate the individual wires about 1½ inches, and strip the insulation back on each wire *no more* than 1/8 inch.

Position the expansion unit motherboard with the foil side up and solder a wire of the cable to each of the 50 pins at P3. No bare wire can be exposed; do not melt the insulation. Slide the ribbon cable through the slotted opening in the expansion unit and fasten the motherboard into place.

If you choose not to cut the slots in the panels, put two 25-pin female connector shells on this end of the cable also. You can then slide these connectors onto P3 of the expansion unit motherboard after it is fastened into place. If you go this route, don't forget to order the extra connector shells and spring clips. With that, the enclosure is complete.

Installation

Installation is simple. Make sure the #14 wire pair that ties the two +8 volt supplies is connected. Plug the H8 into the ac socket of the expansion unit and place the H8 power switch in the on position, where it will remain permanently. Make sure the expansion unit is off.

Insert the expansion PC board onto P3 of the H8 motherboard. Move the lowest addressed memory board onto a slot of the expansion unit motherboard. One end of the interconnecting cable should already be connected to the expansion unit. Route the cable through both slots and insert the connector shell end onto the header strips of the expansion PC board. Do not twist the cable around.

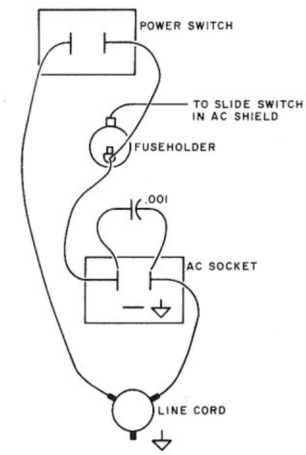


Fig. 3. Wiring aid for ac modification in expansion unit.

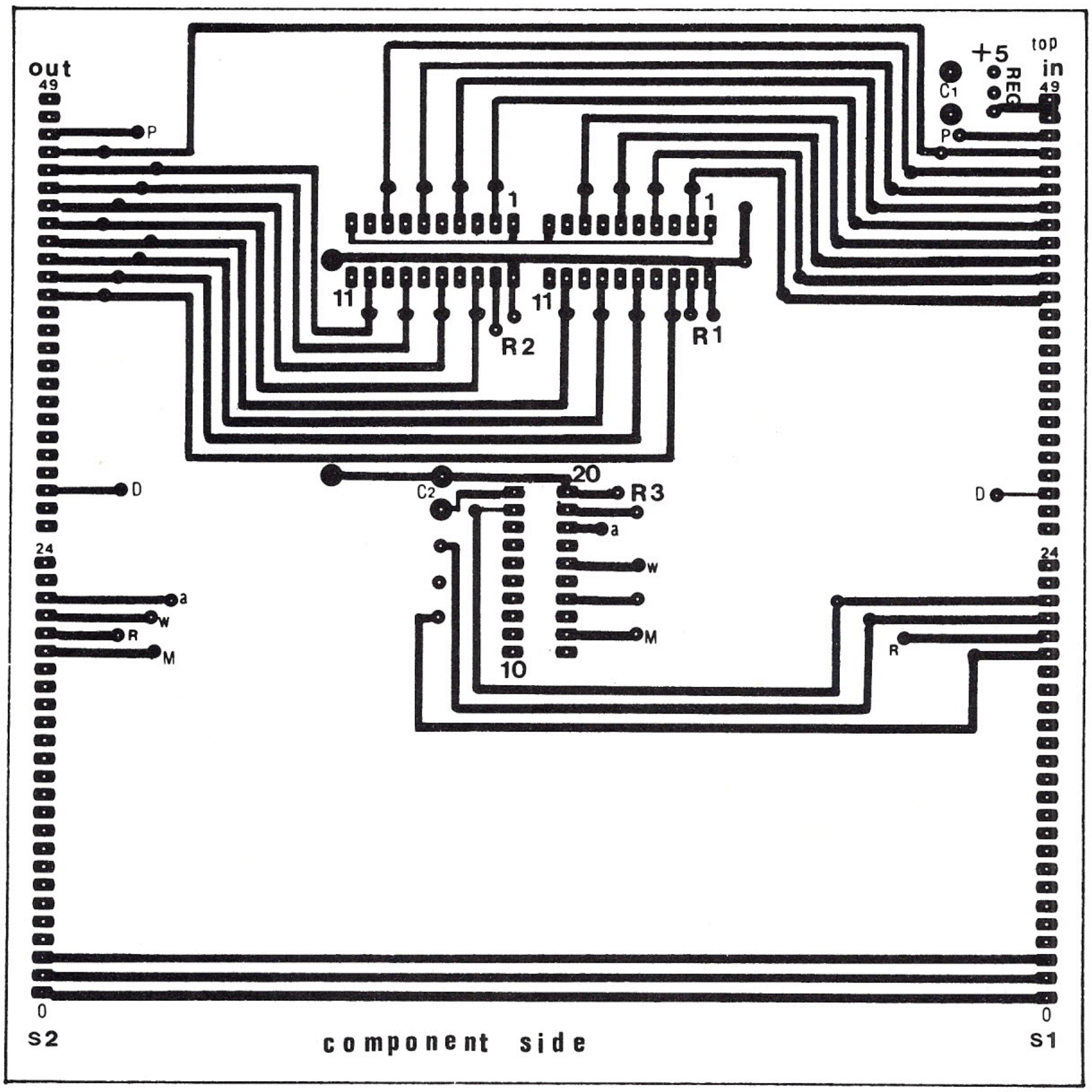


Fig. 2a. Component (top) side of PC board layout.

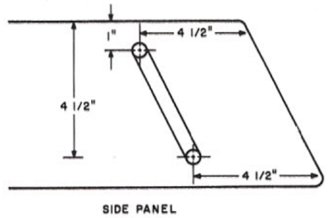


Fig. 4. Mechanical diagram for slotted openings in side panels.

Turn the expansion unit on. If the ION, RUN and PWR LEDS come on and you hear the familiar beep, the board is working and the two units are communicating. This is the reason you use the lowest-addressed memory board for the test, since the computer cannot access any memory if it cannot communicate with the first 8K board.

If any one of the three LEDs do not light, a problem exists. Turn the expansion unit off and remove the expansion PC board from the H8. The relatively simple design of this board should make the problem easy to locate. Check for solder bridges and unsoldered connections and test all parts and connections for continuity. If everything is OK, you may have a bad 74LS241. Locate the problem, then check the fuse in each unit before re-testing. One of them may have blown during the initial test.

As a final test, bootstrap the system and run a commonly used BASIC program that resides on disk. This will fully test the communications between the two units. We experienced no problems at all and were running the system with a full 56K before we knew it.

Conclusion

Our unit has been in operation for months. Now we wonder how we got along with only 32K.

To complete the expansion unit, we put a translucent red cover over the rectangular hole (such as is on the H8) and an opaque black cover over the keypad hole. We are also planning to add an LED digital clock to be located behind the red cover to be used for timings.

Overall, we think the Heath H8 system is excellent. Heath supports its equipment. The technicians at the Heath computer lab can provide us with sufficient assistance over the telephone to correct the few hardware problems we have had. The assembly documentation for all kits is outstanding, as are the user's manuals.

Heath has done well in supplying software, considering how few companies provide any outside support for this system. However, Lifeboat Associates (Suite 505, 2248 Broadway, New York NY 10024) recently released a version of CP/M for the Heath. Apparently this versatile operating system will support high-level language compilers. Lifeboat has already released a version of COBOL to operate with their CP/M.

The Heath System is a pleasure to own, and with the addition of the expansion unit, you can make it more so. ■

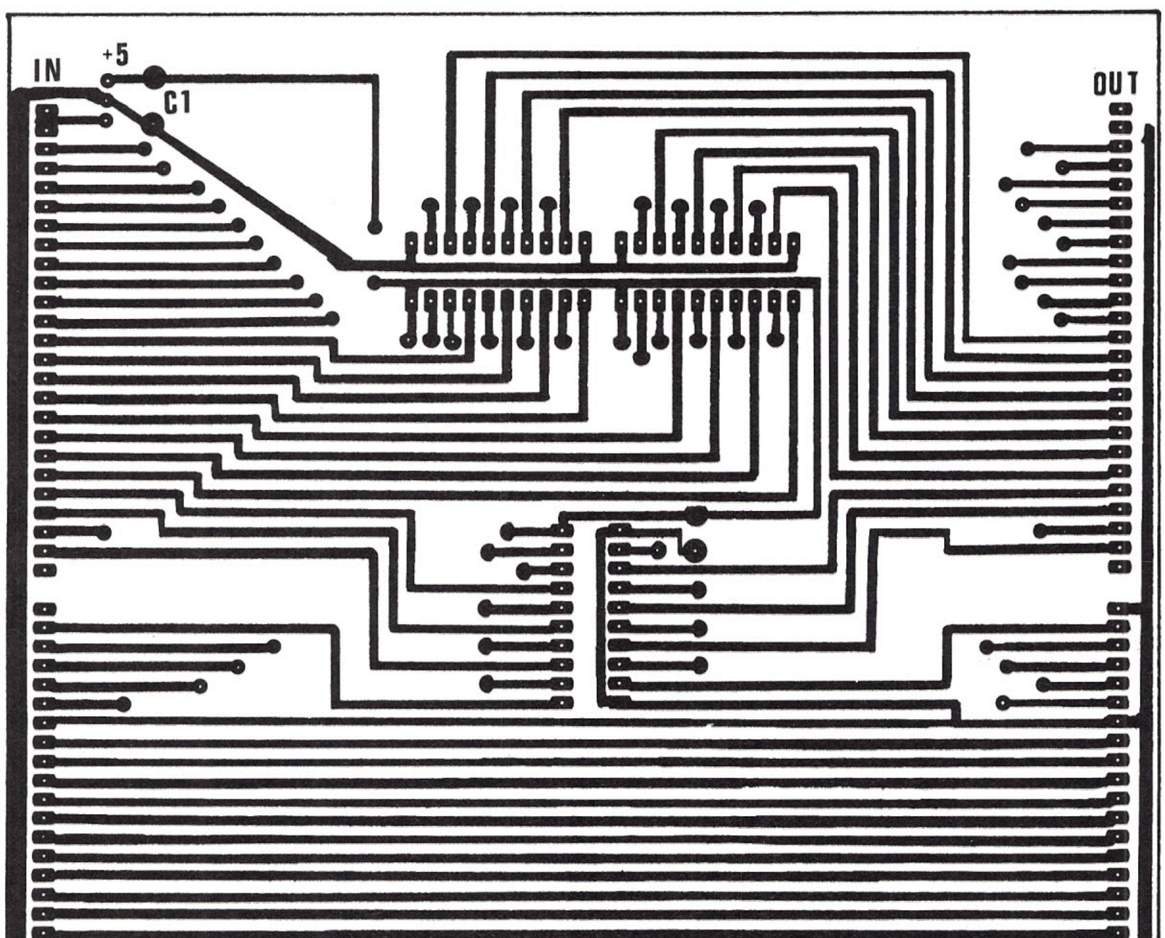


Fig. 2b. Bottom side of PC board layout.

The New HP-41C

Is it a calculator or a computer? Read on and see what you think.

Gregory R. Glau
PO Box 1627
Prescott AZ 86302

What's a review of a calculator doing in a computer magazine?

While the new Hewlett-Packard HP-41C is a hand-held calculator, it shares many of the same features of its larger brother, the microcomputer. In fact, you might say the new HP-41C is a micro/mini/macro handheld computer!

Features

The HP-41C ($295) can handle all the operations you'd expect in a small calculator (sine, cosine, tangent, radians, pi, square root), but it can also use and display alphanumeric strings. You can store and recall strings and even compare them.

It is fully programmable and has continuous memory, which means your program information and data are saved when the calculator is turned off. This allows you to use the alphanumeric display capability to ask for input (SALES ?) and to label output (PROFIT = %). This hand-held unit, like a microcomputer, can talk back to you. The HP-41C will produce ten different beep tones under software control.

The HP-41C also continuously stores three catalogs of information. The first lists by name the programs you have stored in memory, so you always know exactly what programs of your own are available to you at any one time. The second catalog lists the programs contained in any application module you have plugged into the unit. This catalog will read out 0 when you don't have a module working with the unit. The third catalog lists all the operations the HP-41C can perform . . . all 130 of them.

The calculator also works with a list of error messages, which helps in debugging your programs. You can step through a program line-by-line in either direction. The calculator also has a weak-battery indicator (the four "N" batteries will last about a year).

The HP-41C automatically inserts blank program lines into your program as you write it, similar to writing on a small computer in BASIC and skipping lines by tens to leave room for more program lines. If you've left out any lines, you can insert them into the program; the rest of your program will be bumped down the appropriate number of spaces. Likewise, when you've put extra lines into the program, you can delete them, and the rest of the program memory will be moved up.

Besides this full editing control, once you've completed your program, you can execute a PACK function, which removes all the extra lines and makes the program run faster.

Space-Saving Techniques

The HP-41C doesn't use a GOSUB function; instead, you use an execute command, for example, XEQ 30. Then, at whatever line your subroutine starts, you write LBL 30 (label 30).

So, as your program grows (and they all seem to), the line starting your subroutine might be bumped down over and over until, for instance, it ends up at line 45. Line 45 would then read 45 LBL 30. When the subroutine is called for (XEQ 30), it doesn't go to line 30, but to LBL 30. Your programs can grow without having to change all your subroutine numbers and locations.

Using the HP-41C, you assign a name to every program (HEAT, NET, SALES). At the end of that program, you insert an END statement to let the unit know that the program ends at that point in memory. You can then follow in memory with another program (assuming you have the memory space) headed by the program name.

You can then call up any program by name and run it. Contrast this to the old programmable calculator technology that allowed you to input only one program at a time into the unit. Now, you can have as many programs in memory as you have room for and execute them individually.

Although the programs are isolated from each other, the subroutines are not. This means you can access any subroutine from any program.

You can imagine the extra space and time you gain because of this feature. You could, for example, store five financial programs you regularly use. While they all might handle a different task, perhaps they'd use a common subroutine, which you could place in any of the programs and access and execute it from any of the others.

If one of your programs does go outside of itself to get to a subroutine, it might take a second or two for it to search and find it. But the next time you run the program, the HP-41C will remember where that subroutine was and spend no time searching for it!

You can also isolate a subroutine inside a program (have your program search only in the current program for the subroutine). The HP-41C will execute up to six levels of subroutines.

Memory space and program space are

The HP-41C—more than a calculator.

shared. When it's first turned on, the calculator has 17 registers for data (or alphanumeric) storage and 46 registers for program information (which can handle up to 300 lines). You can change this ratio at any time (even while you're programming). There also are additional plug-in modules available, so you could—with four modules hooked up to the HP-41C—end up with five times the original capacity for storage and/or program memory.

To use the alphanumeric capability of the HP-41C, you have to push the ALPHA switch on the calculator. This allows you to talk to the unit by using the letters (which are also printed on the keyboard) of the alphabet.

You can also use the ALPHA mode to execute functions not listed on the keyboard. The standard keyboard shows 58 functions, but the HP-41C can perform 130 different functions.

To make the calculator stop briefly in a running program—to display data or the result of some calculation—you must execute this function, since there's no PAUSE function on the keyboard.

First, you must push the XEQ key (execute). Then, you must push the ALPHA key to allow you to talk to the HP-41C using the letters of the alphabet. Then, key in PSE (the code for PAUSE). Finally, hit the ALPHA key once again to remove the calculator from ALPHA mode.

All this sounds much more complicated than it really is. If you were to put a PAUSE command into a program, your keystrokes would be XEQ ALPHA PSE ALPHA—not really complex at all. You can perform any function on the HP-41C by executing (XEQ) it.

Hewlett-Packard has carried this one step further by adding a user mode to the HP-41C. This allows you to reassign any calculator function to any key.

For example, you might want to replace a seldom used function (cosine) with another function (PAUSE, for instance). To do so you'd simply put the calculator into the user mode, push the COS key . . . and it would automatically put a PAUSE into your program. HP even gives you two overlays and some stickers to put on them so that you can keep track of user-assigned keys. It's convenient to be able to execute a function or operation with just a couple of keystrokes—better than keying in the function alphabetically. But I found that the user mode is most valuable during program execution.

For example, if you have five programs stored—HEAT, SOLAR, PROFIT, TIME and NET—and each performs a task you need to handle every day, the continuous memory of the HP-41C keeps your programs ready to go whenever you want them.

To run a program, you must push XEQ, then the ALPHA key, then key in the program name and then the ALPHA key again. While this example only requires a few keystrokes (XEQ, ALPHA, HEAT, ALPHA), if you run a program ten times—each time inputting different data—you would quickly wear out your key-pushing finger.

Hewlett-Packard has made it possible to assign a program to any key. Once you've assigned a program to a key (the same way you assigned PAUSE to the COS key), you can run things much faster. Instead of pushing, for example, XEQ, ALPHA, SOLAR, ALPHA to run the SOLAR program (eight keystrokes), you can simply push one key.

However, the function you "assign over" is not lost. If you assigned the HEAT program, for instance, to the 1/x key, the HEAT program will run when you push the 1/x key while the calculator is in the user mode (the USER switch has been pushed). But if you hit that USER switch again—to put the calculator back where you started—the 1/x function will again become available. And the HP-41C remembers your assigned keys while it's turned off!

So, in effect, you could completely customize the calculator to make instantly available those functions you use all the time. In short, you can design the calculator yourself!

Extras

Available options include a printer ($350), a card reader ($195) to store program and data information and extra memory modules ($45). Hewlett-Packard is working on a wand to plug into the HP-41C to read printed bar codes directly into program memory.

Although a limited amount of software—most of it from the older HP67/97 software—is available, more is on the way . . . designed just for the special capabilities of the HP-41C, both in the form of application books ($12.50) and plug-in modules ($45). The new calculator comes with a detailed instruction book (267 pages).

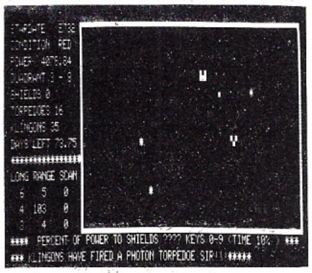
Conclusion

Like everything else, the HP-41C is limited. You only get one readout at a time; while on a small computer you can fill the video screen with data. It also lacks the speed of a microcomputer.

One salesman told me that the HP-41C, with four memory modules plugged in, had about 4K of usable capacity. Not bad for something you can hold in your hand and carry home with you, out on the job or on a plane trip. It sure beats lugging around a bulky microcomputer.

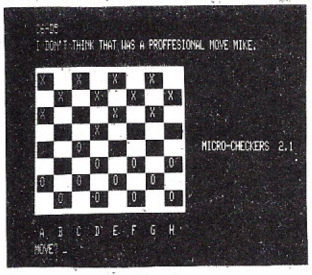
It won't do everything a small computer will—and it wasn't intended to—but it comes close. ■

SOFTWARE → TRS-80 ← SOFTWARE



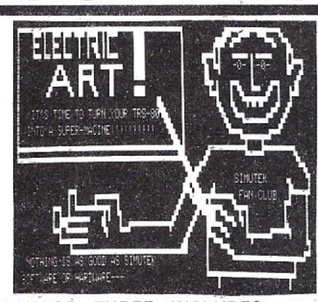
PACKAGE ONE INCLUDES: GRAPHIC-TREK "2000" — This full graphics, real time game is full of fast, exciting action! Exploding photon torpedoes and phasers fill the screen! You must actually navigate the enterprise to dock with the giant space stations as well as to avoid klingon torpedoes! Has shields, galactic memory readout, damage reports, long range sensors, etc! Has 3 levels for beginning, average, or expert players! ★ **INVASION WORG** — Time: 3099, Place: Earth's Solar System Mission: As general of Earth's forces, your job is to stop the Worg Invasion and destroy their outposts on Mars, Venus, Saturn, Neptune, etc! Earth's Forces: Androids — Space Fighters — Lazer Cannon — Neutrino Blasters! Worg Forces: Robots — Saucers — Disintegrators — Proton Destroyers! Multi level game lets you advance to a more complicated game as you get better! ★ **STAR WARS** — Manuever your space fighter deep into the nucleus of the Death Star! Drop your bomb, then escape via the only exit. This graphics game is really fun! May the Force be with you! ★ **SPACE TARGET** — Shoot at enemy Ships with your missiles. If they eject in a parachute, capture them — or if you're cruel, destroy them! Full graphics, real time game! ★ **SAUCERS** — This fast action graphics game has a time limit! Can you be the commander to win the distinguished cross! Requires split second timing to win! Watch out!

ONLY $12.95



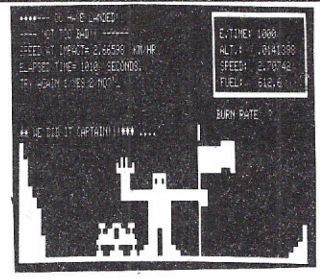
PACKAGE TWO INCLUDES: CHECKERS 2.1 — Finally! A checkers program that will challenge everyone! Expert as well as amateur! Uses 3-ply tree search to find best possible move. Picks randomly between equal moves to assure you of never having identical games. ★ **POKER FACE** — The computer uses psychology as well as logic to try and beat you at poker. Cards are displayed using TRS-80's full graphics. Computer raises, calls, and sometimes even folds! Great practice for your Saturday night poker match! (Plays 5 card draw). ★ **PSYCHIC** — Tell the computer a little about yourself and he'll predict things about you, you won't believe! A real mind bender! Great amusement for parties. ★ **TANGLE MANIA** — Try and force your opponent into an immobile position. But watch out, they're doing the same to you! This graphics game is for 2 people and has been used to end stupid arguments. (And occasionally starts them!) ★ **WORD SCRAMBLE** — This game is for two or more people. One person inputs a word to the computer while the others look away. The computer scrambles the word, then keeps track of wrong guesses.

ONLY $12.95



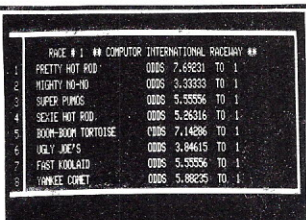
PACKAGE THREE INCLUDES: POETRY — This program lets you choose the subject as well as the mood of the poem you want. You give TRS-80 certain nouns or names, then the mood, and it does the rest! It has a 1000-word + vocabulary of nouns, verbs, adjectives and adverbs! ★ **ELECTRIC ARTIST** — Manual: draw, erase, move as well as, Auto: draw, erase and move. Uses graphics bits not bytes. Saves drawing on tape or disk! ★ **GALACTIC BATTLE** — The Swineus enemy have long range phasers but cannot travel at warp speed! You can, but only have short range phasers! Can you blitzkrieg the enemy without getting destroyed! Full graphics — real time! ★ **WORD MANIA** — Can you guess the computer's words using your human intuitive and logical abilities? You'll need to, to beat the computer! ★ **AIR COMMAND** — Battle the Kamikaze pilots. Requires split second timing. This is a FAST action arcade game.

ONLY $12.95



PACKAGE FOUR INCLUDES: LIFE — This Z-80 machine language program uses full graphics! Over 100 generations per minute make it truly animated! You make your starting pattern, the computer does the rest! Program can be stopped and changes made! Watch it grow! ★ **SPACE LANDER** — This full graphics simulator lets you pick what planet, asteroid or moon you wish to land on! Has 3 skill levels that make it fun for everyone. ★ **GREED II** — Multi-level game is fun and challenging! Beat the computer at this dice game using your knowledge of odds and luck! Computer keeps track of his winnings and yours. Quick fast action. This game is not easy! ★ **THE PHARAOH** — Rule the ancient city of Alexandria! Buy or sell land. Keep your people from revolting! Stop the rampaging rats. Requires a true political personality to become good! ★ **ROBOT HUNTER** — A group of renegade robots have escaped and are spotted in an old ghost town on Mars! Your job as "Robot Hunter" is to destroy the pirate machines before they kill any more settlers! Exciting! Challenging! Full graphics!

ONLY $12.95



PACKAGE FIVE INCLUDES: SUPER HORSERACE — Make your bets just like at the real racetrack! 8 horses race in this spectacular graphic display! Up to 9 people can play! Uses real odds but has that element of chance you see in real life! Keeps track of everyone's winnings and losses. This is one of the few computer simulations that can actually get a room of people cheering! ★ **MAZE MOUSE** — The mouse with a mind! The computer generates random mazes of whatever size you specify, then searches for a way out! The second time, he'll always go fastest route! A true display of artificial intelligence! Full graphics, mazes & mouses! ★ **AMOEBA KILLER** — You command a one man submarine that has been shrunken to the size of bacteria in this exciting graphic adventure! Injected into the president's bloodstream, your mission is to destroy the deadly amoeba infection ravaging his body! ★ **LOGIC** — This popular game is based on Mastermind but utilizes tactics that make it more exciting and challenging — has 2 levels of play to make it fun for everyone. ★ **SUBMARINER** — Shoot torpedoes at the enemy ships to get points. Fast action graphics, arcade type game is exciting and fun for everybody!

ONLY $12.95

HARDWARE → TRS-80 ← HARDWARE

MICRO SPEED

Upgrade your "slow" TRS-80 to a **SUPER FAST MACHINE!!** (2.66 MHZ) over **50% FASTER!** Some of the features:

Auto turn-off during cassette or disk access. (This means NO lost programs EVER!) (Turns back on automatically too!) MANUAL control. (Unit may be turned on or off at any time. Yes even during program execution!) Keyboard indicator light "blinks" when micro-speed is on. Stops blinking when off! **Don't wait for SARGON II or any other program!!!** Comes with easy to follow instructions. (Some soldering required.) **OR** take to your local computer store or TV-Appliance Center for quick installation. (5-10 minutes!!) Works with any model, TRS-80.

ONLY $24.95 complete

MICRO BEEP

Simple hook up: Just plug cassette remote jack into unit.

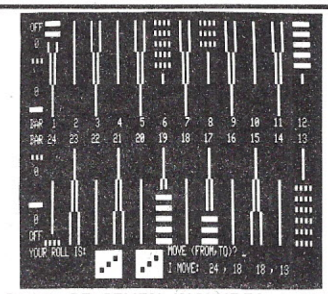
EASILY CONTROLLED FROM BASIC:

OUT 255,4 = on
OUT 255,0 = off

MICRO-BEEP make games more fun as well as provide useful sound output for professional applications!

Works with Any Model I TRS-80

ONLY $9.95 complete



PACKAGE SEVEN INCLUDES: BACKGAMMON 5.0 — 2 different skill levels make this game a challenge to average or advanced players FAST (15 second avg) Looks for best possible move to beat you! FANTASTIC GRAPHICS. Plays doubles and uses international rules ★ **SPEED READING** — Increases your reading speed. Also checks for comprehension of material. Great for teenagers and adults to improve reading skills ★ **PT 109** — Drop depth charges on moving subs. Lower depths get higher points in this fast action graphics game. ★ **YAHTZEE** — Play Yahtzee with the computer. This popular game is even more fun and challenging against a TRS-80! ★ **WALL STREET** — Can you turn your $50,000 into a million dollars? That's the object of this great game Simulates an actual stock market!

ONLY $12.95

PACKAGE SIX INCLUDES: 20 HOME FINANCIAL PROGRAMS — Figures amortization, annuities, description rates, interest tables, earned interest on savings and much, much more. These programs will get used again and again. A must for the conscientious, inflation minded person.

ONLY $12.95

SIMUTEK

✓12

Exceptional Products through Research & Imagination

Send Check, Money Order or Bank Card No. orders to:

SIMUTEK
P.O. Box 35298
Tucson, AZ 85740

Please Add 2.50
Per Order For
Postage & Handling

Master Charge

24 HOUR (7 days) HOTLINE
(602) 882-3948
(C.O.D. $3 extra)

Visa

Same Day Shipment on Bank Cards, Money Orders & C.O.D.

All Tape Programs Require a Minimum of 16K Level 2
Packages Available on Diskette (32K System) $4.25 Extra

3 or More Packages Get 10% Discount

Dealer Inquiries Invited

TRS-80 IS A REGISTERED TRADEMARK OF TANDY CORP.

Data Communication Error Measurement

This 8080 machine-language program checks for errors in data-transmission systems.

Terry A. Conboy
1231 Crestview Drive
San Carlos CA 94070

One of the more fascinating aspects of computing is data communications. Extensive networks link together commercial computers with satellite processors, remote terminals and even other large machines. The promise of electronic mail cannot be realized without an intricate web of data communication channels.

Computer hobbyists are only just beginning to see the potential of communication with the machines of other enthusiasts for bulletin boards, computer game competition, expanded processing capabilities and other imaginative uses.

Common to all users of data communication, whether commercial or hobbyist, is the need to maintain the reliability of the transmission channels and data communications equipment. Developers of modems must be able to verify the performance of their designs in some quantitative way. The technique presented here will allow the hobby computer user to make the measurements necessary to perform both of these functions.

There is a large variety of commercial test equipment available to check data communications links. Almost all of the units carry price tags larger than the total investment by an average hobbyist. Fortunately, there is a relatively simple way to generate sequences of bits that can be inspected by a digital circuit or computer to ferret out errors caused by noise, distortion, interference or other mischief in the transmission path.

My first thought was to build a simple version of a standalone test box with the capability of sending and checking a stream of bits for errors in transmission. Having almost completed the design of this error-rate test box, I figured that it would be worthwhile to parallel the development of the hardware with a program for my computer system to perform the same function. It didn't take long to realize that the software solution was far easier to accomplish, and I soon completely forgot about the hardware approach.

Error-Checking Methods

Several approaches could be taken to measure the error performance of a data communications system. Use of a cyclic redundancy check (CRC) is not specific enough. It lets you know that there is an error in a block transmission, but there could easily be more than one.

A parity bit for each character could be generated and inspected. This has a few problems. Multiple errors can easily cancel out. If the transmission channel is really in bad shape, such as is often the case on HF radio circuits, characters may be missed entirely, along with the parity bit that is supposed to check them. Certain codes, such as the five-level code (Murray), do not provide for parity bits at all.

The "quick brown fox" that has been jumping over lazy dogs for many decades could be used to spot erroneous characters. Again, multiple errors per character could not be easily detected. It would also seem that synchronization at the receiving end could be a problem with the fox test sequence.

Alternating characters containing reversing bit patterns, such as the R-Y test used in five level, would allow bit error checking, but could fail to pinpoint errors that occur when other characters are transmitted.

By far the most common method, and the one with the fewest negative points, is the use of a *pseudorandom binary sequence (PRBS)*. The PRBS can be generated and checked in hardware or software. It allows an accurate count of errors at the receiving end. Because of the variety of patterns of bits created, any sensitivity of the data communication equipment to particular patterns will be stimulated.

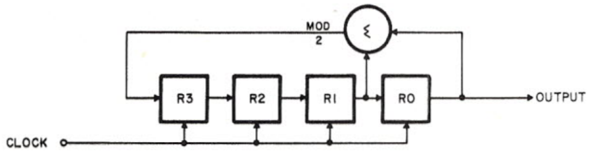
Generating a PRBS

The first item to consider is the method used to generate a PRBS. The usual approach is to use a long shift register with multiple feedback taps. The feedback taps are added *modulo 2*, and the result is stuffed back into the input of the shift register. This is shown for a four-stage shift register in Fig. 1.

There is no mystery to modulo 2 arithmetic. The result of a modulo 2 addition is obtained by adding the numbers in the normal fashion and then dividing by 2. The remainder is the sum mod 2. In a binary numbering system, this is really a snap. Just add all of the single bits (one-digit binary numbers) and ignore the carry. For adding only two bits, this is identical to the exclusive OR function. Still another way of generating the mod 2 sum is to take the odd parity of the bits to be added. If there are an odd number of 1s, the result is equal to 1. If there are an even number of 1s, the answer is 0.

For the four-stage shift register shown, there are fifteen different states. You might expect there would be sixteen, which is the number of different combinations of four bits. Unfortunately, when all of the registers contain 0, the PRBS generator will latch up, since the feedback will always be 0. Whether the sequence is generated in software or hardware, this all 0 condition must be checked for to avoid initialization problems.

In general, the number of states that can occur until the sequence repeats is $(2^n - 1)$, if the proper taps are chosen for the (n) stages. Table 1 indicates the positions of the taps and the sequence length for shift registers up to sixteen stages. There are other combinations of taps that will produce maximal-length sequences, but those given involve the minimum number of taps. "Pseudorandom Sequences and Arrays" (F. Jessie



STATE	REGISTER CONTENTS			
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	1	0	0	1
4	1	1	0	0
5	0	1	1	0
6	1	0	1	1
7	0	1	0	1
8	1	0	1	0
9	1	1	0	1
10	1	1	1	0
11	1	1	1	1
12	0	1	1	1
13	0	0	1	1
14	0	0	0	1
0	1	0	0	0
1	0	1	0	0
.	.	.	.	.

Fig. 1. A four-stage pseudorandom binary sequence generator. The contents of the registers change state on one edge of the clock. In this case, the modulo 2 summer could be replaced by an exclusive OR gate.

MacWilliams and Neil J.A. Sloane, *Proceedings of the IEEE*, December 1976, p. 1715) gives a complete discussion of PRBS generators and an extensive bibliography.

Checking for Errors

Although the sequences generated by the feedback shift registers can be extremely long, they are also very predictable. This is what makes them useful for error checking. My first reaction to the concept made me wonder if it was necessary for the error detection logic to scan the entire sequence before the detector could "find its place."

Fortunately, this isn't the case. The detector circuit must only receive a number of correct bits equal to the number of stages in the PRBS generator before it can predict exactly which binary value the next bit should be. If there is a disagreement, the new bit is in error.

An error detector for a PRBS generated by a four-bit shift register is shown in Fig. 2. The error detector must have the same number of stages as the generator. It must also have the same configuration of feedback taps. The difference between the generator and the error detector is that there is no actual feedback in the detector. The modulo 2 sum of the feedback taps is merely compared to the bit, which will be shifted into the register next. This should be the same bit that was shifted into the generator shift register several time states previously.

The exclusive OR gate functions as a data comparator. If the incoming bit differs from the sum of the taps, the XOR gate output will go to a 1, indicating an error.

Remember that the detector shift register must be full of correct bits before an erroneous bit can be detected. Complications arise when the bad bit is shifted into the shift register on the following clock pulses. Eventually this bit gets to the taps and causes an incorrect sum to be formed. The result is that the new incoming bit is declared to be wrong even though it is actually right. For isolated bit errors, there will be an error noted for each tap, plus the one for the first time it is found. In the case of the four-stage detector in Fig. 2, there will be three errors counted for each bad bit received.

This becomes further complicated if the errors occur in bursts. You can imagine what will happen if another error comes along just at the same time a previous error is in position at the taps. Two wrongs seem to make a right. In practice this will cause the error total to be slightly smaller than you would have expected. Unless the data communications channel that you are monitoring is really rotten, the underestimate of errors won't be large.

Use in Asynchronous Systems

What we have been considering is an oversimplified data transmission system. There has been no mention of the source of the clocks for the two shift registers. Obviously, the clock at the transmitting end sets the baud rate for the system. The clock at the receiving end must be magically regenerated from the transitions of the incoming bits or must be sent in some other devious fashion from the transmitting end, perhaps by a separate channel. Regeneration of the received clock is a normal function in a synchronous data transmission arrangement, but not many hobbyists are using synchronous systems. For that reason, we'll restrict our attention to asynchronous schemes.

At first glance, using an asynchronous transmission system might seem to present some problems. What do you do about the required start bit, parity bit (if used) and stop bit(s)? Fortunately, normal methods of sending serial data take care of all of these details. A device such as a UART (universal asynchronous receiver-transmitter) is typically used in a serial I/O port.

At the transmitting end, the desired number of the least significant bits of the PRBS shift register are parallel loaded into the UART for transmission. The UART handles the addition of the overhead bits required for transmission. At the receiving end, it takes care of character-by-character synchronization and removes all of the added bits. The parallel output of the UART is then shifted into the error detector one bit at a time. This can be done at any convenient rate as long as it is fast enough to be completed before the next character arrives.

Put another way, we pretend that groups of bits from the PRBS make up the information bits of asynchronous characters. We end up sending the bit sequence a bunch at a time without mixing up the original order.

A note of caution is worth mentioning: even though the sequence of bits from the shift register at the sending end is

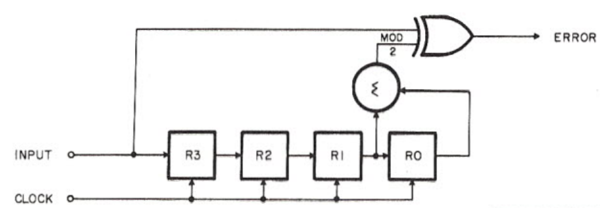


Fig. 2. A four-stage PRBS error-checking circuit. There is no actual feedback as in the generator. The modulo 2 sum of the taps is compared with the incoming bit by the exclusive OR gate. An output of 1 indicates an error. Here again, the summer could be an exclusive OR gate.

Number of Stages	Feedback Taps On Registers	Sequence Length
2	1,0	3
3	1,0	7
4	1,0	15
5	2,0	31
6	1,0	63
7	1,0	127
8	6,5,1,0	255
9	4,0	511
10	3,0	1023
11	2,0	2047
12	7,4,3,0	4095
13	4,3,1,0	8191
14	12,11,1,0	16383
15	1,0	32767
16	5,3,2,0	65535

Table 1. Required feedback taps and resulting pseudorandom binary sequence length for shift registers from two to 16 stages.

Program listing.

```
0000 C5    COMTST  PUSH B      Save registers on stack
0001 D5            PUSH D
0002 E5            PUSH H
0003 F5            PUSH PSW
```

pseudorandom, the stream of characters that results is not really random, as might be judged by certain mathematical standards. This isn't the best way to implement a random number generator for gambling purposes.

Program Description

The error measurement program is written in 8080 machine language. I'm sure that with suitable cleverness, it could be done in BASIC, but the shift and rotate instructions of the processor are ideally suited to the generation and checking of the bit sequences. Why make it hard?

A 15-stage configuration was chosen for the PRBS generator and error detector. This provides a long sequence without creating a software mess. A 16-stage design would have required the simulation of two more taps and would have complicated the error-checking arrangement. The varied 16-bit manipulations in the 8080 instruction set simplify the program greatly.

A flowchart of the program appears in Fig. 3. There are three main sections in the program. The control section takes care of initialization, reset of the error count and exit from the routine. The transmission section implements the PRBS shift register and sends the individual characters to the modem I/O port. The receiving section gets characters from the modem I/O port, shifts the bits through the error-checking shift register and keeps and finally outputs the total of any errors found.

By far the most confusing aspects of the program are the PRBS generator and the error-checking section. Both the transmit and receive shift register contents are kept in RAM between generation and checking of the bits in a character.

In the transmitting section of the program, the shift register contents are transferred to the HL register pair. There is a check for all 0s to avoid a latch-up state. The contents of the L register are then moved to the accumulator, and all but the two least significant bits are masked off. The parity flag is then checked. If the parity is odd, then the feedback is 1. To stuff the feedback into the shift register, the H register is ORed with 80 hex to set the most significant bit to 1. This step is skipped when the feedback is 0.

After the feedback is added, the H and L registers are rotated (in the accumulator) through the carry bit. This permits the bit that falls out of the H register to be moved into the L register. This calculation of feedback and shifting to the right is repeated once for each of the information bits to be sent in the character (7 for ASCII).

The contents of the HL regis-

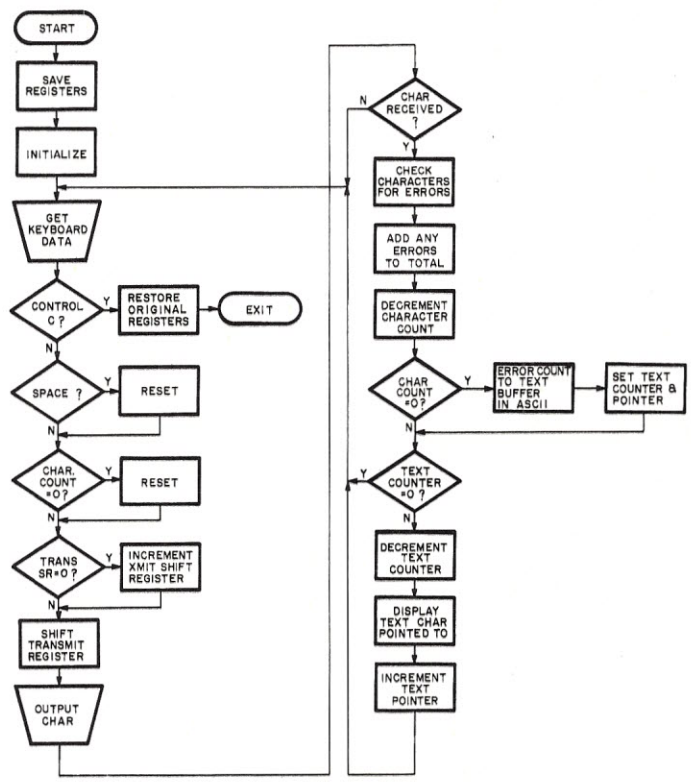


Fig. 3. The flowchart of the error-rate measurement program.

```
0004  AF                XRA A
0005  32 05 01          STA TXTCNT   Zero Text Counter
0008  CD D8 00          CALL RESET
000B  DB 03    CONTRL   IN KBSTAT    Get Keyboard Status
000D  E6 02             ANI 02H      Bit 1 = Data Available
000F  CA 25 00          JZ CKCNT     If no data, bypass decoding
0012  DB 02             IN KBDATA    Get Keyboard Data
0014  E6 7F             ANI 7FH      Ignore any Parity Bit
0016  FE 03             CPI 03H      Is it "CONTROL-C"?
0018  C2 20 00          JNZ CONT     No, continue program
001B  F1                POP PSW      Yes, return registers
001C  E1                POP H
001D  D1                POP D
001E  C1                POP B
001F  C9                RET          and exit...
0020  FE 20    CONT     CPI 20H      Is it a "SPACE"?
0022  CC D8 00          CZ RESET     Yes, reset Error & Char Counters
0025  2A 06 01 CKCNT    LHLD CHCNT   Get Character Counter from RAM
0028  7C                MOV A,H
0029  B5                ORA L        Is Counter zero?
002A  CC D8 00          CZ RESET     Yes, reset Error & Char Counters
002D  0E 07             MVI C,07H    Set up no. of Information Bits
002F  2A 08 01          LHLD TSR     Get Transmit Shift Reg from RAM
0032  7C                MOV A,H
0033  B5                ORA L        Shift Reg contents all zeroes?
0034  C2 38 00          JNZ OK       No, do nothing
0037  23                INX H        Yes, bump register to unlatch
0038  7D       OK       MOV A,L
0039  E6 03             ANI 03H      Select Feedback Taps
003B  7C                MOV A,H
003C  EA 41 00          JPE FBZ      If Parity Even, Feedback is zero
003F  F6 80             ORI 80H      If Odd, set Feedback to one
0041  1F       FBZ      RAR          Shift Feedback into higher bits
0042  67                MOV H,A
0043  7D                MOV A,L      Transfers shift out via Carry
0044  1F                RAR          Shift lower order bits
0045  6F                MOV L,A
0046  0D                DCR C        Decrement Info Bit Counter
0047  C2 38 00          JNZ OK       Loop until all bits shifted
004A  22 08 01          SHLD TSR     Keep Shift Reg contents in RAM
004D  DB 01    CHSEND   IN MDSTAT    Get Modem UART Transmit Status
004F  E6                ANI 01H      Bit 0 = Transmit Buffer Empty
0051  CA 4D 00          JZ CHSEND    Wait until empty
0054  7D                MOV A,L      Get low order bits of Shift Reg
0055  D3 00             OUT MODEM    Send the new Character formed
0057  0E 07             MVI C,07H    Set up no. of information bits
0059  06 00             MVI B,00H    Clear Char Error Counter
005B  2A 0A 01          LHLD RSR     Get Rcvr Shift Reg contents
005E  DB 01             IN MDSTAT    Get Modem UART Receive Status
0060  E6 02             ANI 02H      Bit 1 = Receive Data Available
0062  CA 0B 00          JZ CONTRL    Go to pgm beginning if no char
0065  DB 00             IN MODEM     Get new Received Character
0067  00                NOP          Room to output the character
0068  00                NOP           to your front panel...
0069  0F       AGN      RRC          Rotate LSB to MSB
006A  57                MOV D,A      Save rotated character
006B  E6 80             ANI 80H      Mask off all but MSB
006D  5F                MOV E,A      Save incoming bit
006E  B4                ORA H        Combine w/MSD of Rcv Shift Reg
006F  67                MOV H,A
0070  7D                MOV A,L      Get LSD of Rcv Shift Reg
0071  E6 03             ANI 03H      Select Feedback Taps
0073  B3                ORA E        Combine with incoming bit
0074  EA 7C 00          JPE GOOD     If Parity even, Bit is correct
0077  3E 01             MVI A,01H    If odd, set A to 01
0079  80                ADD B         and add to Char Error Count
007A  27                DAA            in BCD
007B  47                MOV B,A
007C  7C       GOOD     MOV A,H
007D  1F                RAR          Shift higher order bits
007E  67                MOV H,A
007F  7D                MOV A,L
0080  1F                RAR          Shift lower order bits
0081  6F                MOV L,A
0082  7A                MOV A,D      Restore rotated Rcvd Character
0083  0D                DCR C
0084  C2 69 00          JNZ AGN      Repeat until all bits checked
0087  22 0A 01          SHLD RSR     Restore Rcv Shift Reg in RAM
```

ter pair are moved back to RAM, and the required least significant bits are sent as a serial character. Actually, eight bits are sent to the UART, and the way it is configured determines how many bits will be used in the character sent.

In the receiving section of the program, the least significant bit of the received character, the bit to be checked, is moved into the most significant bit of the E register. After the two least significant bits of the shift register contents are isolated by masking, the E register is ORed with the accumulator to leave an eight-bit number with only three active bits. These three bits represent the two feedback taps and the incoming bit that is being checked. If the parity of the accumulator contents is even, the bit is good. If the parity is odd, the received bit is bad and the error count for that character is incremented (in BCD).

After each bit is checked, the receiving shift register, contained in the HL register, is shifted. The checking and shifting are repeated until all of the information bits are checked. Then the error total is updated.

The error count is output as "BER/10↑4 = XXXX." This is a cryptic way of saying that the bit error rate per 10^4 bits is XXXX. In actuality, 10,000 bits were not really checked. Remember that each error going through the receiving shift register produces three counts. For this reason, only 3333 bits need to be checked. For ASCII characters with seven information bits, this represents 476 characters. In hexadecimal this is 1DC. This is the value to which the character count is reset for each measurement cycle.

To run the program, you must call it from your operating system or another program. The program saves all of the 8080 registers and restores them when the test program is terminated. The only assumption made by the program is that the stack pointer has been properly initialized. Space for at least 14 bytes on the stack must be available.

Pressing the space bar on your keyboard during execution will reset the error total and initiate a new measuring cycle. This is useful to clear the count after the data path has been interrupted. This includes any time that the program is called, since the receiving register will take two or three characters to properly synchronize with the incoming PRBS.

Port	Direction	Function of Port	Status Bit Information
00	Out	Data to Modem	
00	In	Data from Modem	
01	In	Status of Port 00	b0-Output Buffer Empty;b1-Input Data Ready
02	In	Data from Keyboard	
03	In	Status of Port 02	b1-Keyboard Data Ready

Table 2. Input/output port assignments used in the program. Use of status bits is indicated.

Note that the program will continue to send characters as long as it is executing. The error count will be output only during times that characters are being received. If the program is to be used for receive-only measurements, such as in half-duplex data systems or tape applications, the transmit serial output can be ignored.

To terminate the operation of the test program, type "Control-C" (the end-of-text character). The routine will "pop" all of the original contents of the 8080 registers and will return to the calling program.

Program Modifications

It is extremely unlikely that the program as listed will run on anybody's system but mine. My system is Z-80 based and has two RS-232C serial I/O ports. One of them is used to interface a serial keyboard. The other is used to connect to a modem or other communications device. The measurement of error rate is displayed via a DMA video interface using a standard driver routine that is part of my system monitor program.

The addresses of the I/O ports in my system and the associated status bits are given in Table 2. Besides modifying the I/O assignments, the only other change that should be necessary is to supply the address of the routine that you use to send data to your CRT. Although my CRT is driven by a shared-memory interface, any output device is usable. The output device must be at least as fast as the system being tested. For instance, a 1200 baud CRT terminal would do nicely if the data

```
008A  2A 0C 01          LHLD ERRCNT  Get error total from RAM
008D  78                MOV A,B      Contains Char Error Count
008E  85                ADD L        Add LSD Error Counter
008F  27                DAA           in BCD
0090  6F                MOV L,A
0091  3E 00             MVI A,00H    Clear A but not Carry
0093  8C                ADC H        Add Carry & MSD of Error Count
0094  27                DAA           in BCD
0095  67                MOV H,A
0096  22 0C 01          SHLD ERRCNT  Return Error Total to RAM
0099  2A 06 01          LHLD CHCNT   Get Character Counter from RAM
009C  2B                DCX H        Decrement it
009D  22 06 01          SHLD CHCNT   And return to RAM
00A0  7C                MOV A,H
00A1  B5                ORA L        Check it for zero

00A2  C2 BF 00          JNZ DSPE     If not zero, bypass buffer load
00A5  21 00 01          LXI H,TBFR   Set up pointer to Text Buffer
00A8  3A 0D 01          LDA HIERCT   Get high byte of Error Count
00AB  CD E5 00          CALL BCDS    Hex to ASCII & into Buffer
00AE  3A 0C 01          LDA ERRCNT   Get low byte of Error Count
00B1  CD E5 00          CALL BCDS    Hex to ASCII & into Buffer
00B4  3E 0F             MVI A,0FH    Initialize Buffer Counter
00B6  32 05 01          STA TXTCNT    and keep in RAM
00B9  21 F5 00          LXI H,MSG    Set pointer to message area
00BC  22 0E 01          SHLD TXTPTR   and keep in RAM
00BF  3A 05 01  DSPE    LDA TXTCNT   Get Text Counter from RAM
00C2  B7                ORA A        Check for zero
00C3  CA 0B 00          JZ CONTRL    If no more, back to beginning
00C6  3D                DCR A        If more text, reduce Counter
00C7  32 05 01          STA TXTCNT    and put back in RAM
00CA  2A 0E 01          LHLD TXTPTR  Get Text Pointer
00CD  7E                MOV A,M      Get Message Char pointed to
00CE  CD 03 F8          CALL CHD      and have it displayed on CRT
00D1  23                INX H        Bump pointer value
00D2  22 0E 01          SHLD TXTPTR   and return it to RAM
00D5  C3 0B 00          JMP CONTRL   Go back to beginning for more
00D8  21 DC 01  RESET   LXI H,01DCH  No. of char per measurement
00DB  22 06 01          SHLD CHCNT    to initial value in Counter
00DE  21 00 00          LXI H,0000H  Zero value
00E1  22 0C 01          SHLD ERRCNT   put into Error Counter
00E4  C9                RET
00E5  47        BCDS    MOV B,A      Temporarily hold number
00E6  0F                RRC          Rotate
00E7  0F                RRC           to
00E8  0F                RRC            get the
00E9  0F                RRC             higher hex digit
00EA  CD EE 00          CALL ASCII   Convert hex digit to ASCII
00ED  78                MOV A,B      Now do lower hex digit
00EE  E6 0F     ASCII   ANI 0FH      Select lower four bits
00F0  F6 30             ORI 30H      Make it an ASCII numeral
00F2  77                MOV M,A      Place in buffer area
00F3  23                INX H        Bump pointer value
00F4  C9                RET
00F5  0D        MSG     ASCII        Carriage Return
00F6  0A                ASCII        Line Feed
00F7  42                ASCII        'B'
00F8  45                ASCII        'E'
00F9  52                ASCII        'R'
00FA  2F                ASCII        '/'
00FB  31                ASCII        '1'
00FC  30                ASCII        '0'
00FD  5E                ASCII        '↑'
00FE  34                ASCII        '4'
00FF  3D                ASCII        '='

0100  30        TBFR    ASCII        '0' Will be changed by the
0101  30                ASCII        '0'  program to reflect the
0102  30                ASCII        '0'   number of errors found
0103  30                ASCII        '0'    in the measuring cycle
0104  00                NOP
0105  00        TXTCNT  DATA         Keeps count of text left to go
0106  DC 01     CHCNT   DATA         Keeps count of char left to go
0108  55 55     TSR     DATA         Contents of Transmit Shift Reg
010A  55 55     RSR     DATA         Contents of Receive Shift Reg
010C  00        ERRCNT  DATA         Low byte of Error Count Total
010D  00        HIERCT  DATA         High byte of Error Count Total
010E  F5 00     TXTPTR  DATA         Pointer to Message & Text Bufr
0110  00                NOP
```

Number of Information Bits Used	Constant at 002E and 0047	Two-byte constant at 00D9-00DA for an equivalent measurement of about				Time for a 10^4 bit measurement cycle seconds (with bps)
		10^3 Bits	10^4 Bits	10^5 Bits	10^6 Bits	
5	05	0043	029B	1A0B	FFFF*	110 (45.5)
6	06	0038	022C	15B4	D904	41 (110)
7	07	0030	01DC	129A	BA03	16 (300)
8	08	0024	01A1	1047	A2C3	3.5 (1200)

* Will give results about 1.7 percent too low. Length of measurement period is over three hours at 45.5 bps!

Table 3. Hexadecimal constants to be changed to give proper operation with a different number of information bits and with different measuring intervals. Approximate measurement cycle times are given for various data rates. Constants used in program are indicated in boxes.

link being measured were running at 1200 baud or less.

In addition to modifying the program to fit your system, you may want to change the equivalent number of bits over which the error rate is measured. For systems which are virtually error-free, a greater number of bits should be checked to give a higher probability of finding erroneous bits. Fewer bits per measuring cycle are desirable when system reliability is poor or to shorten the cycle time when making adjustments to improve performance.

Table 3 gives the constants that must be changed in the program to vary the number of bits per measurement cycle. Values for characters with five to eight information bits are included. An indication of the cycle time is given for some typical data rates. The values in boxes are those from the original program.

Applications

A common use of the program is to send test sequences over a loopback. Several different arrangements are shown in Fig. 4. Note that the loopback may be for digital signals, such as RS-232, directly at the computer or at the interface to the terminal equipment at the other end, or it may be for analog signals at either the sending modem or at the far end of the transmission path.

Some commerical modems have the capability of being looped-back at various points under control from a remote location. This permits maintenance to be done without requiring manual patching of signals. The part of the system causing a problem can easily be isolated with this technique.

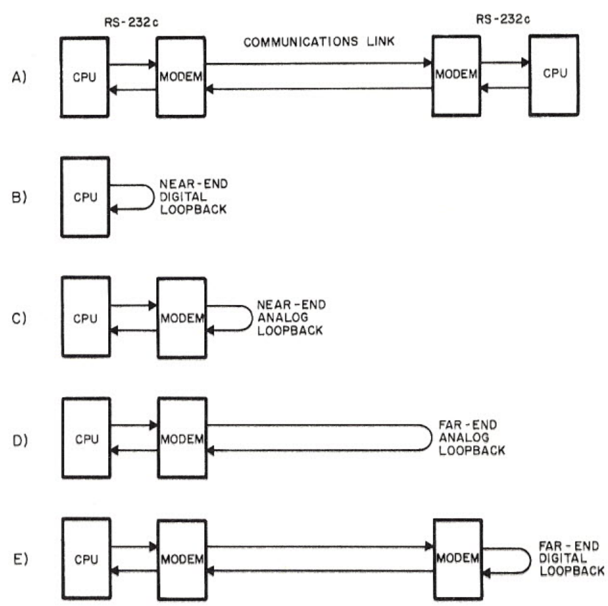


Fig. 4. Equipment arrangements for testing with the program. With the program executing in two communicating computers, configuration a can be used. Testing from one end only is possible with digital or analog loopbacks at various points as in b, c, d or e.

Of course, tests between two computers can be done if both of the processors are executing the program at the same time. One advantage of this arrangement is that it permits finding sources of errors that occur for transmissions in one direction only. These maybe found by loopback tests, but you will have trouble deciding which half of the loop is the culprit. Testing between two computers is also necessary for systems that are half duplex by nature. Radio channels in the HF spectrum are normally only one direction at a time on the same frequency and must be tested this way.

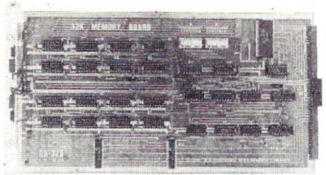
Another possible use of error-rate testing is to permit adjusting tape interfaces for data recording. The PRBS is first recorded and is then played back to allow counting errors that may be occurring.

Final Comments

I hope that this technique for measuring errors is helpful to hobby users of data communications. Even if you think my code is inefficient (which is probably the case), you should be able to use the ideas that are presented, even on other processors.

This method is another simple trick that the commercial people take for granted. As a result, hobbyists never find out the secrets that are "obvious" to the pros. ■

HEATH® COMPATIBLE



DG-32D 32K RAM FEATURES:

- Plugs into Heath® H8 Computer.
- Ready to use. Fully assembled, tested & burned in.
- Operates with existing Heath® memory.
- Protected memory Output Buffers in the event of Address error.
- Utilizes popular 4116 RAM devices.
- Memory Address DIP switch changeable.
- Arranged as 4 Independent 8K Blocks.
- Low Power Consumption: Less than 6 watts, typical.
- Transparent Refresh.
- One year guarantee.
- Compatible with all current H8 peripherals.

D•G Electronic Developments Co. brings you a totally compatible, fully assembled and tested 32K RAM for Heath® H8 computers. The DG-32D has less than 6 watts power consumption. This allows you to add a full 32K bytes of Random Access Memory without taxing or replacing your computer's power supply. Engineered to plug-in and run without any user modifications, the DG-32D can be used with or without existing H8 RAM without modification. Protection of the memory output buffers is provided in the event of assigning two blocks to the same address space. The DG-32D is the ideal answer to expansion of the Heath® H8 computer. . . Low power consumption, low price, high capacity, total engineering and exacting production methods.

Heath® and H8 are registered trademarks of the Heath Corporation, Benton Harbor, Michigan.

BURNED IN 16K MEMORY CHIP SET
H88-H89 . . . . **$89.00**

ORDERING INFORMATION: DG-32D RAM available only from DG Electronic Developments Co., P.O. Box 1124, 1827 South Armstrong, Denison, Texas 75020. Check, money-order, VISA or Master Charge. Phone orders accepted on charge orders. NO COD's. Foreign orders add 30%. Texas residents add 5%. For VISA or Master Charge orders call 214-465-7805. $479.00 freight prepaid. Allow three weeks for personal checks to clear banks.

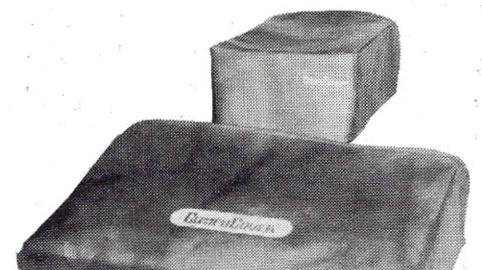
D•G ELECTRONIC DEVELOPMENTS CO. ✓157

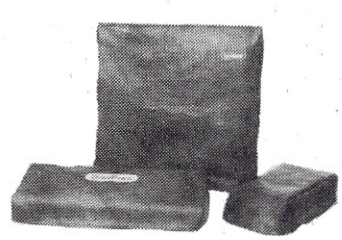
32K RAM $479.00

1st Anniversary Sale

CompuCover® ✓90

Thank You For Our 1st Year





10% Discount
On All Retail Orders
Postmarked Before May 31st
That Mention This Ad.

- Cloth Backed Naugalhyde Vinyl
- Waterproof & Dustproof
- Longer Life
- Improved Reliability
- Three Decorator Colors

Saddle Tan • Electra Blue • Black

TO ORDER:
Include $1.00 for
Postage & Handling
Send Check or Money Order To
CompuCover
P.O. Box 324 Dept. A
Mary Esther, FL 32569
Phone (904) 243-5793

TRS-80 MODEL II
Entire Unit $22.95
Keyboard Only 7.95

TRS-80 MODEL I
Keyboard $7.95
Cassette 4.95
Video Display 9.95*
Package Offer . . . 19.95*
*Note - Add $3.00 for EXPANSION INTER.

Full Apple II
Cover $12.95
Apple II Keyboard . 9.95
Apple II Disk 3.95

Pet $15.95
Sorcerer 9.95

Hazeltine
1400, 1410, 1500, 1510, 1520,1552 . . . . . . $18.95

Soroc IQ 120 . . . . . 18.95
Adds Terminal
980, 100, etc 19.95
ADM-3 . . . . 14.95
General Automation
3380 . . . . 19.95
Honeywell VIP
7200 . . . . 22.95
IBM 5251
Display Station . . . 22.95

Data General
Keyboard $7.95
Data General
Video 15.95

Wang Terminal
without disk $18.95
Wang 2221 Printer 19.95

TRS-80
Line Printer I . . . $17.95
TRS-80
Line Printer II 9.95

Centronics
701, 703 . . . . 19.95
Centronics 730 . . . . 9.95
Centronics 779 . . . 17.95
T.I.
800, 810, 820, 825. 19.95
Trendcom 100, 200 . 9.95
Decwriter III
36, 120, 180 . . . . . . 19.95
Integral Data Systems
440 . . . . 12.95

TRS 80 Disk $4.95
Percom Disk 4.95
Double Disk 7.95
Lobo Disk
5¼" mini 6.95
Matchless Disk
5¼" mini 6.95
Vista Disk
5¼" mini 6.95

Dial-up Directory

Our dedicated dialer tells why the microcomputer is a practical communications device.

Frank J. Derfler, Jr.
PO Box 17283
Montgomery AL 36117

This month we will describe the practical aspects of using the home computer as a communications device. We will focus on one outstanding free service and explain some technical terms in simple form.

Time Tyranny

The U.S. has the most elaborate communications system in the world. Our tightly wired web of communications includes television, radio, cable TV, private carrier systems and the extensive telephone system. The major inconvenience in plugging into this system is that you have to be ready to receive when a message of interest is sent. You have to arrange your life around the time set for the evening news, or you will miss it. You have to be home to receive a phone call, or you will not get it.

Now, a cresting wave of consumer products is helping us break free of the time tyranny of telecommunications. Video recorders, telephone answering machines, portable telephones and mini tape recorders make the messages of the communications system available to us when we want them, not just when the messages are being sent. While you and your family go out to dinner, one tape recorder can record the evening news and another can answer the telephone. Later, at your leisure, you can check to see if the communications system has any messages for you.

This is the idea behind several different kinds of computer communications services available to you today. These services store information until you are ready to see it. A computer serves as a smart recorder for you, sorting and cataloging data in its memory for easy retrieval. The types of inputs into the computer, the amount of processing and the degree of personal service available generally separate the types of systems.

Personal hobby systems are variously called Computer Bulletin Board Systems (CBBS), Apple Bulletin Board Systems (ABBS), TRS-80 Forums and other similar names. Hobby systems started out as the computer equivalent of simple public bulletin boards, where notices to individuals and the public could be posted. They are now increasing in the amount of personal service they provide at no cost. They often give regular users special sign-on notices and allow quick sorting of messages by categories. Some systems can also store and transfer programs.

Commercial services available to the home user rival any government or private capability. The Source and Micronet are two of these systems. We'll review both of these in an upcoming article.

Spotlight

Listings are always top-heavy with California systems, but one is tops in flexibility, creativity and reliability: the People's Message System (714-449-5689). Bill Blue is the main person in this people's system, which has all of the standard, and many special, features.

The Apple II Plus that runs this system is dwarfed by its peripherals. The two standard 5-inch drives are augmented by three 8-inch disks and internal ROMs. A clock card, D. C. Hayes Micromodem and local printer complete the package. This large storage capacity provides lengthy "articles" for discussion and comment. It also serves as a program upload and download service for Apple users and others with the right software.

If you type "GENERAL" in response to this system's prompt, you will be in the program mode. You can run programs or select upload/download. This option allows you to store programs in the system for others or to get programs for yourself. The function is designed for Apple users, but instructions are provided for anyone who can save files.

Bill's other special features include Message Alert, Reverse Scan, Flagging and four levels of message security. Message Alert (command M) checks message headers for the name you logged in. It lists all messages with your name in them. This is done automatically at log-on also. Reverse Scan lets you look at the summary of messages from the most recent to those first received. This is a real timesaver! You can use a Scan (S) command to specify a message number higher than the highest number in file.

Flagging is done during a scan. When you see a message you want, just send a control R. At the end of the scan, send an asterisk. The flagged messages will come automatically. The message security system determines who can read and kill specific messages.

A special article is often placed on the system to stimulate discussion.

Much comment was recently generated on a lengthy piece describing experiments in telepathy. This activity turns a bulletin system into a kind of computer magazine with instant reader response. I believe I perceive an iceberg tip on the horizon.

The quality and features of Bill's system pay dividends. Some of the most knowledgeable and interesting computer hackers frequent the People's System.

Terms

When you use your computer to communicate over phone lines, you may run into some new terms that often are associated with switch or software options. If you don't get the option right, your machine won't talk.

One important term is duplex. As a communications system planner, I thought full duplex meant that the device could transmit and receive at the same time; half duplex meant the device could either receive or transmit, not both, at any one moment. This applies only for a communications link such as a phone line, radio circuit or microwave shot, but it is not true for computer devices.

When computer designers talk about duplex modes, they bring in new words such as echo-back and echo-plex. Echo means a playback of what a computer received from a terminal.

When a terminal is operating in the full-duplex (echo-plex) mode, sent characters go from the keyboard, out the RS-232 port and over the transmission media (channel) to a computer or host device. They are then echoed back to the terminal and displayed on the screen. This provides a continuous verification of the communications path. Note that it is not exactly the simultaneous operation you think of in a full-duplex radio link. Full duplex is the common mode of remote operation for bulletin board systems.

In computer terminals, half duplex means that the character is displayed and transmitted simultaneously. If your terminal thinks it is in half duplex and your host thinks you are in full, you will see everything you send twice. Each character will appear once when you strike the key and again when it is echoed.

When both computer and remote terminal are operating in half duplex they are transmitting "in the blind." The channel could be lost, but they would keep sending. You can lose a lot of bits that way.

Modems obey the communications rules. Half-duplex modems have to use handshaking to send (RS-232, pin 4) and clear to send (RS-232, pin 5) signal between themselves and their associated terminal or computer. Don't worry about these signals. Full duplex is the common mode of operation.

If you are dialing-up with other individuals, you may run into many different combinations of software, hardware and modems. Software can be particularly confusing because some operating systems echo the screen out the RS-232 port. Other machines echo when in BASIC, but not at other times. It is nearly impossible to make a chart describing all the possibilities.

As a rule, if your computer software or hardware thinks you are in full duplex, start out with the modem in "FULL." If you are getting double characters, flip your modem switch to "HALF." Beyond that, you and the other person will have to experiment with different configurations until you get one that works.

Conclusion

Electronic bulletin systems allow a new use for your computer and an escape from the need to be present to communicate. The price is right, too.

I would like to hear about your good and bad dial-up experiences. Send paper mail (with a stamped envelope, if you want a reply) to PO Box 17283, Montgomery AL 36117. Send electronic mail on the Atlanta Northstar (404-939-1520) or to TCB967 on the Source. ■

The following list provides the location, phone number and other information about dial-up computer communications systems around the country. Note that this list contains the first PET system, the first TRS-80 MOD II and the first Canadian system, as well as creative and practical applications. Unless otherwise shown, all systems operate 24 hours a day. Hit at least three carriage returns to set the speed. I have checked into all of these systems, which were verified from my list of over 120 "reported" systems.

LOCATION	PHONE NO.	COMMENTS
British Columbia		
Vancouver	604-687-2640	Welcome, Canada! CBBS.
California		
Orange County	714-537-7913	Forum-80
Palo Alto	415-493-7691	ABBS
Santa Barbara	805-682-7876	8 AM–11 PM Pacific time
Santa Barbara	805-964-4115	Answers after fourth ring.
Georgia		
Atlanta	404-939-8429	Not full time. ABBS
Iowa		
Iowa City	319-353-6528	5 PM–8 AM Central time. Weekends 24 hours. ABBS. Special-purpose system for medical professionals. Includes medical education programs to run.
Michigan		
Ypsilanti	313-484-0732	PET. Call Fred Hambrecht at 313-485-1896 (voice) for PET software. This system often doesn't use carriage returns.
New Jersey		
Scotch Plains	201-753-1225 201-968-1074	Two systems of the Amateur Computer Group of NJ. This club has 1000-plus members and many special interest and user groups.
New York		
Endicott	607-754-5571	Runs on one North Star disk. Centered more on an individual electronic mail approach. Easy to use and fast. A fine program!
Tennessee		
Nashville	615-254-9193	TRS-80 MOD II. Different software from ACS Services.
Utah		
Logan	801-753-6800	ABBS.

Mickey Gets New Ears

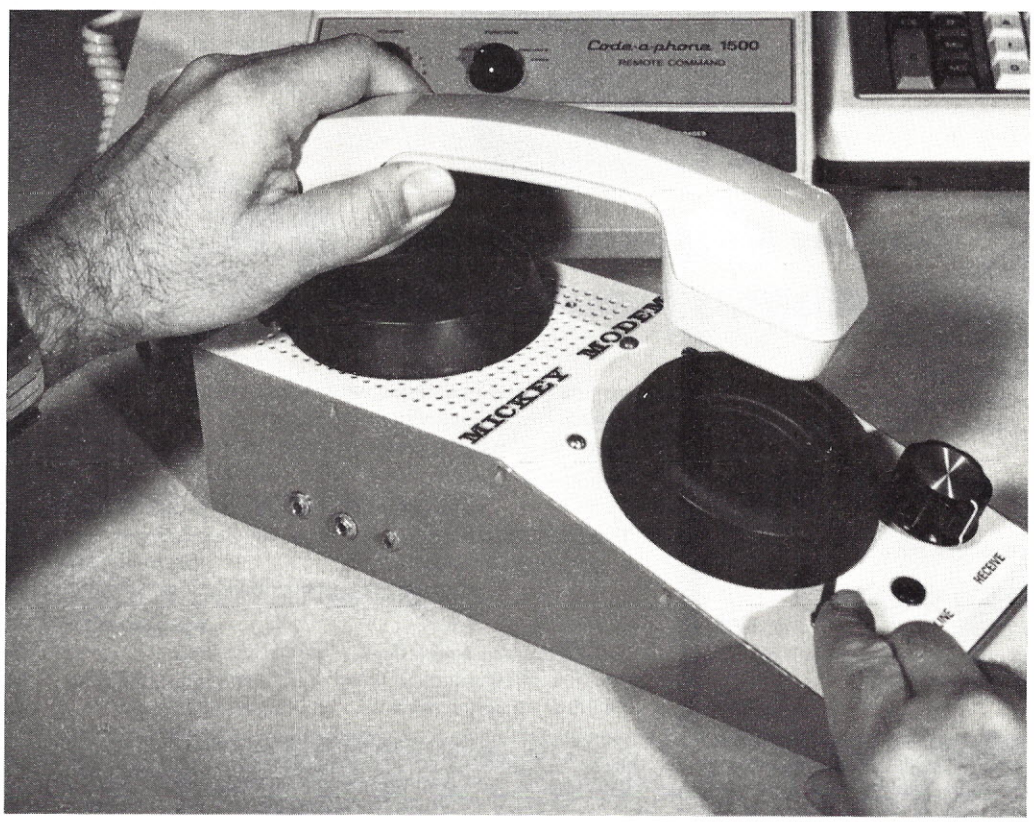
Electronic Systems enhances "Mickey Modem's" "hearing" with its modem board kit.

Stephen Gibson
P O Box 38386
Los Angeles CA 90038

Sending data or whole programs over Ma Bell's lines is easy with Mickey Modem (*Kilobaud,* November 1978, p. 52). It's the only inexpensive way to go as long as both computers are the same or use the same cassette-port formats. Unfortunately, two-way communication with a computer unlike yours is another matter. In fact, it's next to impossible without some common interchange method —a "standard," if you want to call it that. Fortunately for us all, someone else already had this problem, so standards exist. . . several of them!

Nevertheless, the most universal and simplest way to get into operation is the Bell 103 standard. See Fig. 1 for how the Bell 103 ties two computers together. To make the conversion ultra-cheap, I wired the Electronic Systems bare-bones 103 modem board into the original Mickey Modem. The price for the kit (Photo 1) was so low ($27.50) that I bought two modems and implemented both originate and answer capability, thereby ending forever my telephone-interface hassles. This "standard" Mickey appears to be a budget way to beat the high modem price barrier.

The modem kit can be wired for either originate or answer modes. If your plans are to only talk with a time-share computer, then you may only need an originate modem. If you really wanted to keep expenses down, you could use one kit and switch the few frequency-changing components back and forth from originate to answer. I don't recommend it. The extra capacity from wires running everywhere will make your modem drift all over the place! I know; I tried.



Two printed circuit modem kits and a few leftovers from the original Mickey Modem make this originate/answer unit inexpensive and easy to build.

New Circuit Description

All that's involved with the conversion are a few simple modifications to accommodate the new PC board(s). While it may be possible to use the Electronic Systems modem wholly alone, I think the added goodies from Mickey Modem will give you a much better running unit equaling those costing more than $100.

In the clever Electronic Systems design in Fig. 2, the modem board tone receiver uses an XR-2211 phase-lock-loop IC to convert the received frequencies to logic 1s and 0s. To transmit, a 567 phase-lock-loop oscillator is simply shifted in frequency by a single transistor that switches a frequency-changing capacitor in and out of the circuit. Both PLL ICs can be wired for either

	Part	Calculated	Actual
2125 Hz	R1, R2	786 Ohms	820 Ohms
	R3	393 Ohms	390 Ohms
	C1	.2 uF	.15 uF
	C2, C3	.1 uF	.1 uF
1170 Hz	R1, R2	907 Ohms	820 Ohms
	R3	453 Ohms	560 Ohms
	C1	.33 uF	.33 uF
	C2, C3	.15 uF	.15 uF

Table 1. Simple notch filter circuit values.

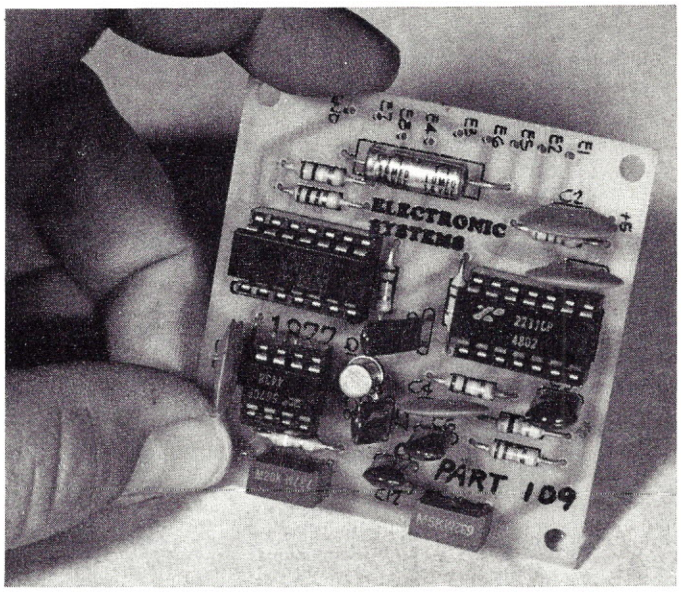


Photo 1. A nifty add-on to Mickey Modem. The Electronic Systems 103 modem kit can be wired for originate or answer frequencies.

originate or answer frequencies.

The circuit diagram to the new standard Mickey is shown in Fig. 3. I chose to acoustically couple the new Mickey to the Ma Bell system. I used a crystal microphone element, which had too low an output to drive the PLL receiver from the line. So I made a trip down to my obsequious Radio Shack store to get another one. After some testing in my lab, I resolved that they didn't have a high-output microphone either. Clearly, I needed some gain... at least enough to drive the modem board. I ended up using one of Mickey's LM386 ICs.

Now the microphone had plenty of level to drive the chip. In fact, I had so much gain that I was able to add a couple of simple RC notch filters to the amplifier output to cancel side-tone frequencies from the modem oscillator. The feedback from the oscillator can sometimes cause the demodulator PLL IC to lock up on the modulator. Clearly, a no-no. The notch filters are each tuned to a midway point between each of the two frequency groups.

Of course, an active notch filter might have been better over this configuration. A hi pass and low pass would work too. But why bother with more ICs and parts when, even with sloppy tuning, you can get 15 to 20 db rejection, which is more than enough? Besides, I wanted to test out the RC notch filter program in the January 1979 issue of *73 Magazine* ("Design-a-Notcher," p.100).

A summary of the values I computed are in Table 1. The nearest EIA values I used are there too. Depending on tolerances, you may have to juggle them just a bit if you wish to hit the notch frequencies exactly.

My modem board didn't have enough output level to work long distances. So I used a 99¢ 500 to 8 Ohm speaker matching transformer. I now had enough level to send solid copy anywhere. That's really saying a lot because my local phone office is the old Hollywood exchange. And like everything else about Hollywood, "It was great in the 30s, but now. . . ."

As a final touch to make Mickey's new ears perk up, I wired a simple carrier-detect LED circuit. If the modem receives a mark frequency (originate or answer mode),

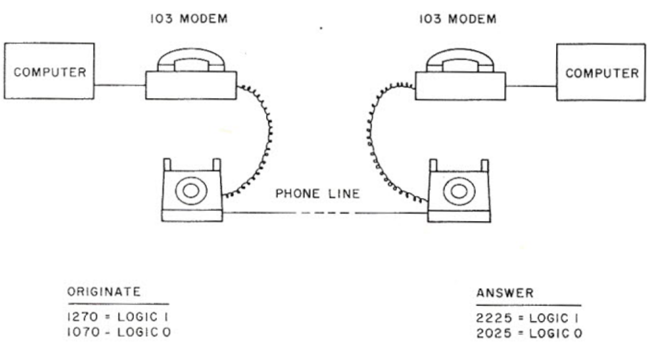


Fig. 1. A Bell 103 setup. Two-way communication is possible because separate frequencies are used to send and receive. Notice that the higher frequency of each pair is a logic 1 (also called a mark).

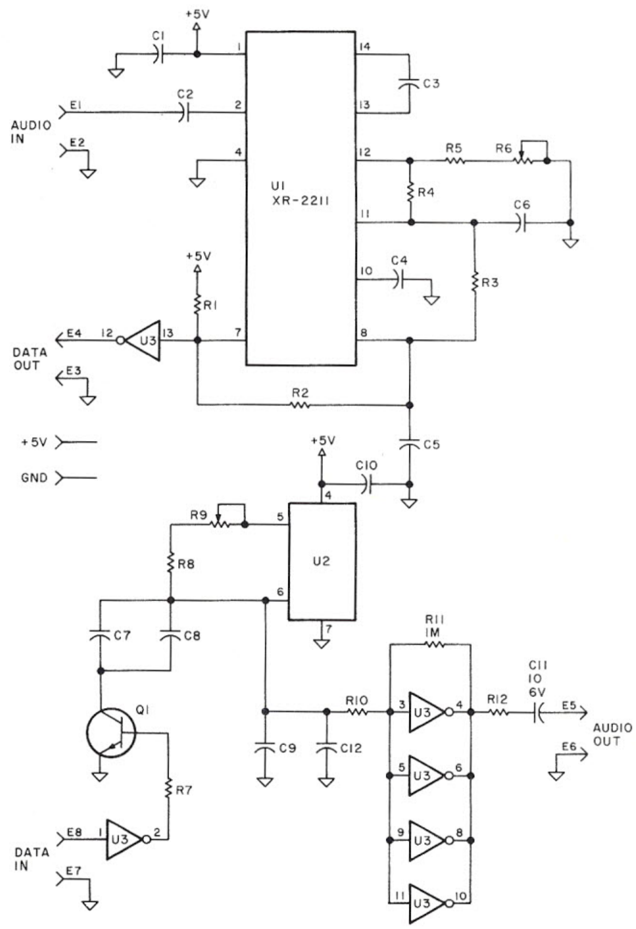


Fig. 2. Use of PLLs makes this circuit easy to tune. Reversing the values of R10 and R11 will increase output.

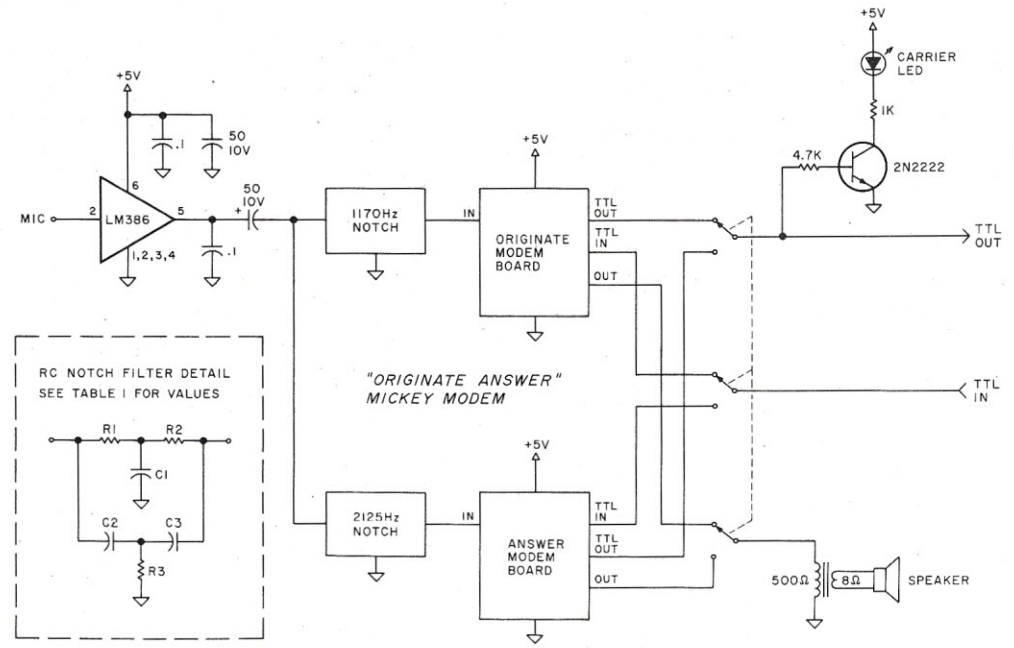


Fig. 3. Modify your Mickey Modem to work computers thousands of miles away with this circuit. An LM386 IC amp boosts the input signal out of the mud to feed 103 Modem kits through noise-reducing filters. A 99 cent transformer ensures a high level to the speaker. The 2N2222 transistor lights the LED to indicate a carrier or mark condition.

which indicates a valid connection, a single 2N2222 transistor will turn on the carrier-detection LED. Pin 5 (unused) of the XR-2211 PLL already has this feature. I opted for the external transistor rather than risk zapping the XR-2211. A three-pole, two-position transfers operation to either modem board.

Whither RS-232?

The inputs and outputs are TTL levels. If you plan to wire your modem directly to the computer and your wires are short, you just might get away with TTL all the way to your mainframe. If the connection is more than a few feet though, you'd be wise to use RS-232 levels. It's difficult to interconnect two units if one is RS-232 and the other isn't. TTL and RS-232 don't connect in a convenient translatable form. In fact, you may damage a TTL device if you connect an RS-232 signal to it.

The reason is simple. TTL levels amount to a logic 1 equaling about 5 volts positive, while a logic 0 is roughly 0 volts. Fine so far. On the other hand, an RS-232 signal logic 1 ranges anywhere from −3 to 9 volts, and a logic 0 is +3 to 9 volts. They almost look backwards, don't they?

Fortunately, you can use a single 1488 and/or 1489 IC to translate either way. Look at Fig. 4. Notice that when you send a TTL logic 1 into the 1488 line driver, you still get a TTL logic 1 out of the 1489 receiver. Neat, huh?. . .especially since you can send RS-232 signals all over the place and not pick up a lot of system-crashing noise.

If your computer serial I/O board has RS-232 levels, you'll need to make the RS-232 conversion back to TTL logic levels. Remember: you must make conversion at each end of the line. You can use the 1488/1489 ICs, or you might want to use the Electronic Systems RS-232 converter kit (see Photo 2), which sells for $7. I mounted mine in Mickey Modem on the edge of one of the modem PC boards.

Putting It Together

The new parts easily fit into an LMB 1042 box with the old Mickey Modem board. A convenient feature about this box is the sloping panel. The cushions for Mickey's ears make a tighter fit to the telephone handset. I used surplus acoustical-coupler cushions, but foam rubber would work just as well. If you can find an acoustical coupler that's still intact, all the better. Less work to mount something usually means less grief.

The modem boards are mounted on a few stand-offs above the speaker (see Photo 3). The LM386 IC amp and the notch filters are mounted on a large Radio Shack experimenter's circuit board. Use shielded wire everywhere. The super gain of the LM386 and the sensitivity of the phase-lock loops dictate careful construction practice.

I used an external power supply, but you could very well sandwich it all into one box. Some parts shifting would be necessary, though it's unlikely you will construct your Mickey Modem exactly like mine anyway.

Construction Tips

Frequency drift can be a problem with this modem, unless you use top-notch capacitors, do a *good* job soldering your PC board connections and use a regulated 5 volt supply. So often I've marveled at the work of PC board solder artists whose work is fantastically beautiful because they use ever so little heat. Yet they scratch their heads in utter amazement when they discover cold solder joints causing their drift problems!

As for regulation, all I can say is that zener diodes are inexpensive. Even the workhorse 7805 IC regulator chip price is within reason, when you consider the frustration you'll likely encounter if your modem drifts.

Direct-Connection Method

A 600/600 Ohm line transformer can be used (see Fig. 5) even though the phone-line impedance is somewhere around 900 Ohms. Fortunately, the few db lost don't upset things too much. Neither does the fact that the modulator and demodulator present dissimilar impedances to the network. If you are a purist, change R12 on the modem board to 620 Ohms and load the input to the LM386 with 620 Ohms.

A better null is possible, but hardly worth the effort when the RC notch filters that follow the amp manage to bury what little is left of the side tone. Rx is used to balance out the transmitted signal from being heard by the receiver. About 20 to 30 db rejection is possible with this arrangement. You can sometimes pick up an extra db or two when Rx is bridged with Cx, which can range from .01 uF to .1 uF.

If you can't find a 600/600 Ohm transformer, try 500/500 or 1000/1000 Ohms. The mismatch is only a few db either way. You

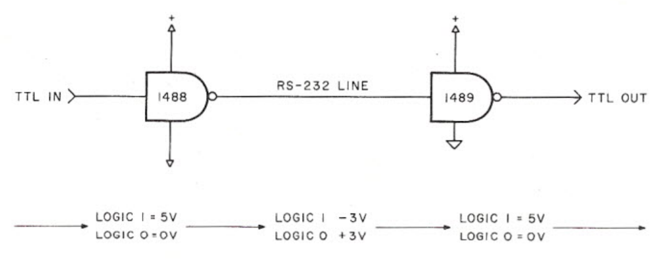


Fig. 4. RS-232 debunked! You can forget about those crazy ± voltages when you use converters. If you send a logic 1 into one end of the line, you should get a logic 1 out the other!

✓Reader Service—see page 258

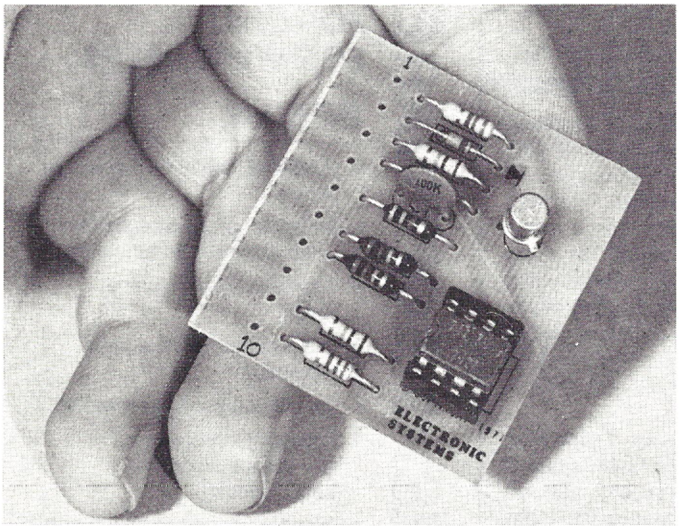
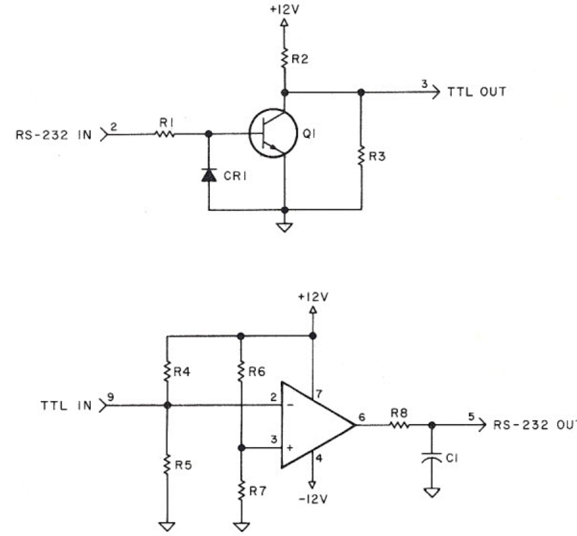


Photo 2. The $7 RS-232 converter board from Electronic Systems.



Circuit for Photo 2.

can even use two 1000/8 Ohm speaker matching transformers wired back to back! By the way, you do know you are supposed to tie your modem to the line via an approved coupler, don't you? I know it seems crazy having your 600 Ohm matching transformer feeding their 600 matching transformer and then the line, but that's the law!

Tune-up Tips

Here's how to tune an originate modem. The exact details for an answer modem are the same; only the frequencies are reversed:

An audio oscillator, scope and frequency counter are best to adjust the modulator section. To use the frequency counter, simply connect it to the modem output and read the frequency while the modem is held at a logic 1 (5 volts TTL or −3 to −9 volts for RS-232). Adjust R9 for 1270 Hz. Switch to a transmitted logic 0 to see that you are sending roughly 1070 Hz. Adjust the size of C7 to get within 5 cycles of 1070 Hz. You could easily rock R9 a few cycles either way and still have the 1270 Hz within the range of a demodulator.

It's possible to get these frequencies "right on" with little effort. Depending on the other modem you may be talking to, you could very well be off by as much as 30 cycles and still have it work. Stay within 5 cycles to be safe though.

If you don't have a frequency counter, you can feed the modem to the scope's vertical input and a good oscillator to the horizontal input (see Fig. 6). When both units are set at the same frequency, you should see a slanted line at about 45 degrees, depending on signal levels, scope gain, etc. Set the oscillator to 1270 Hz and feed a logic 1 into the modem modulator. You might see a square or a bunch of spinning circles. These are called Lissajous figures. Adjust R9 until the figures stop moving; adjust for a single circle. Notice that you can rotate it in apparent three dimensions.

Now gently move R9 until you get a slanted line. This method is as accurate as your oscillator. Now change the oscillator frequency to 1070 Hz and feed the modulator with a logic 0. Adjust the size of C7 until you see another slanted line on the scope.

To adjust the demodulator section, you simply bridge the microphone input with the oscillator set to 2225 Hz (a logic 1) at −30 to −40 dbm. Adjust R6 to light the LED or see a logic 1 on the modem output. Switch the oscillator to 2025 Hz and see that the LED goes out (logic 0).

Ideally, the transition from 1 to 0 should occur midway between 2025 Hz and 2225 Hz, or about 2125 Hz. Jockey R6 and the oscillator around until you achieve a transition at about this frequency. Then you must verify that you still have a logic 1 at 2225 Hz and a 0 at 2025 Hz. Remember: the procedures for an answer modem are the same, but the send and receive frequencies are reversed. Refer back to Fig. 1 for the frequencies.

The last adjustment requires an ac multimeter with a db scale. Check that the tone-send level is no greter than −10 dbm

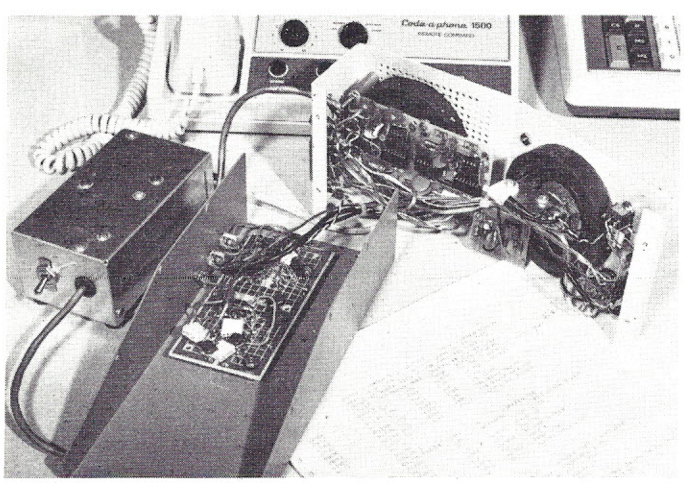


Photo 3. Mickey minus his ears. The microphone amp and notch filters are mounted on the bottom. The two modem boards mount on stand-offs above the speaker. An external power supply was used.

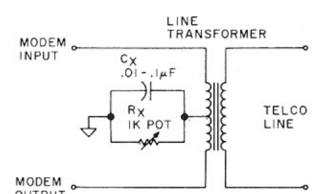


Fig. 5. Direct-connection technique using a line transformer in a hybrid circuit. Adjust Rx to minimize feedback into input.

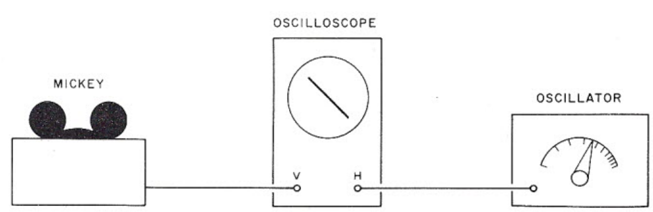


Fig. 6. Equal frequencies from the oscillator and Mickey will display a slant line (45°) on the scope.

Listing 1. A software UART terminal program for an 8080 single-bit input.

```
0001 ;
0002 ;  SOFTWARE UART/MODEM PROGRAM
0003 ;
0004 ;       1978 STEPHEN GIBSON
0005 ;
0010 ;THIS PROGRAM IS A CHEAP AND DIRTY WAY
0015 ;TO GET YOUR MODEM WORKING. IT USES A
0020 ;SINGLE BIT FROM ANY I/O PORT AND DOES
0025 ;SERIAL TO PARALLEL (AND BACK) CONVERSION.
0028 ;THE HEX VALUES FOR BAUD RATES OF 110
0030 ;AND 300 ARE LISTED IN THE TABLE BELOW.
0035 ;THESE RATES ARE FOR AN 8080 2MC CLOCK
0040 ;WITH THE PROGRAM IN RAM. A HALF OR FULL
0045 ;DUPLEX SWITCH OPTION IS TIED TO PORT FF.
0050 ;
0055 TSTAT  EQU   00H           ;TERMINAL STATUS PORT
0060 TDATA  EQU   01H           ;TERMINAL DATA PORT
0065 RMASK  EQU   01H           ;TERMINAL RECEIVE MASK
0070 SMASK  EQU   080H          ;TERMINAL SEND MASK
0075 MMASK  EQU   01H           ;MODEM I/O BIT MASK
0080 DMASK  EQU   080H          ;SENSE SWITCH MASK
0085 DPORT  EQU   0FFH          ;SENSE SWITCH PORT
0090 MPORT  EQU   02H           ;MODEM I/O PORT (BIT 0)
0092 HBAUD  EQU   0056H         ;300 BAUD DELAY VALUE
0095 LBAUD  EQU   00EEH         ;110 BAUD DELAY VALUE
0098 ;
0100        ORG   05000H
0110        LXI   SP,05000H
0115        DI
0120        MVI   C,01          ;GET A 'MARK'
0130        CALL  OUTPT         ;SEND TO MODEM
0140 ;
0150 LOOP   CALL  MODEM         ;CK MODEM STATUS
0160        CALL  TTY           ;CK TTYVATUS
0170        JMP   LOOP          ;KEEP DOING IT
0180 ;
0200 MODEM  LXI   B,0800H       ;SET B TO 8,CLEAR C
0210        IN    MPORT         ;GET MODEM DATA BIT
0220        ANI   MMASK         ;TEST LSB
0230        RNZ                 ;GO CK TERMINAL
0240 ;
0250        CALL  DLY1          ;DELAY 1/2 BIT TIME
0260        IN    MPORT         ;VERIFY START BIT
0270        ANI   MMASK         ;TEST LSB AGAIN
0280        RNZ                 ;MUST BE NOISE
0290 ;
0300 INPUT  CALL  DLY2          ;DELAY 1 BIT
0310        IN    MPORT         ;GET NEXT BIT
0320        ANI   MMASK         ;TEST LSB
0330        ORA   C             ;GET PREVIOUS BITS
0340        RRC                 ;ROTATE 0 TO 7
0350        MOV   C,A           ;SAVE IN C
0360        DCR   B             ;COUNT DOWN 8
0370        JNZ   INPUT         ;GET NEXT BIT
0380        CALL  DLY2          ;DELAY 1 BIT TIME ONLY
0390        CALL  CHARO         ;PRINT ON TERMINAL
0395        RET                 ;GO BACK TO LOOP
0400 ;
0410 TTY    IN    TSTAT         ;CK TERMINAL STATUS
0420        ANI   RMASK         ;TEST LSB
0430        RNZ                 ;GO CK MODEM
0440 ;
0450        IN    TDATA         ;GET DATA BYTE
0470        MOV   C,A           ;STORE IN C
0480        CALL  DUPLX         ;DO WE ECHO?
0490        CALL  OUTPT         ;SEND TO MODEM
0495        RET                 ;GO BACK TO LOOP
0500 ;
0510 DUPLX  IN    DPORT         ;LOOK AT SENSE SW
0520        ANI   DMASK         ;MASK ALL BUT SW 15
0530        RZ                  ;IS IT DUPLEX?
0540        CALL  CHARO         ;MUST BE HALF DUPLEX
0550        RET                 ;BACK TO TTY
0560 ;
0570 OUTPT  MVI   B,8           ;BIT CTR
0580        XRA   A             ;CLEAR A
0590        OUT   MPORT         ;OUTPUT START BIT
0600        CALL  DLY2          ;DELAY 1 BIT TIME
0610        JMP   $             ;30
0620        JMP   $             ;CLOCKS
0625        JMP   $             ;FOR BALANCE
0630        MOV   A,C           ;GET BYTE
0640 BITO   OUT   MPORT         ;OUTPUT BIT
0650        CALL  DLY2          ;DELAY 1 BIT TIME
0660        RRC                 ;ROTATE TO NEXT BIT
0670        NOP                 ;16 CLOCKS
0680        NOP                 ;FOR
0690        NOP                 ;MORE
0700        NOP                 ;BALANCE
0710        DCR   B             ;COUNT DOWN BITS
0720        JNZ   BITO          ;SEND MORE?
0730        MVI   A,01H         ;SET STOP BIT
0740        OUT   MPORT         ;OUTPUT STOP BIT
0750        CALL  DLY2          ;FOR 1 BIT ONLY
0755        CALL  DLY2          ;FOR 1 BIT ONLY
0760        RET                 ;BACK TO TTY
0770 ;
0780 CHARO  IN    TSTAT         ;GET TERMINAL STATUS
0790        ANI   SMASK         ;TEST MSB
0800        JNZ   CHARO         ;LOOP TILL READY
0810        MOV   A,C           ;GET BYTE FROM C
0820        ANI   07FH          ;CLEAN IT UP
0830        OUT   TDATA         ;SEND TO TERMINAL
0840        RET                 ;BACK TO DUPLX OR INPUT
0850 ;
0860 DLY1   PUSH  PSW           ;SAVE A
0870        PUSH  H             ;AND HL
0890        LXI   H,LBAUD       ;110 BAUD
0900        JMP   DLY           ;GO TO IT
0910 ;
0920 DLY2   PUSH  PSW           ;SAVE A
0930        PUSH  H             ;AND HL
0940        LXI   H,LBAUD       ;110 BAUD
0950        DCX   H             ;SLOW
0960        DCX   H             ;DOWN
0970        DAD   H             ;DOUBLE HALF BIT TIME
0980 DLY    MOV   A,H           ;GET MSB
0990        ORA   L             ;GET LSB
1000        JZ    DONE          ;ALL BITS ZERO
1010        NOP                 ;ADD 4 CLOCKS
1020        DCX   H             ;AND THIS ONE 100
1030        JMP   DLY           ;LOOP AGAIN
1040 DONE   POP   H             ;RESTORE HL
1050        POP   PSW           ;AND A REG
1060        RET                 ;EXIT TO MODEM,INPUT,OUTPT,BITO
```

Listing 2. A serial-to-parallel software UART program for the TRS-80. The cassette port is used. A resistor inside the TRS-80 must be removed before you run this program.

```
                00010 ; ***  TERMINAL I/O PROGRAM  ***
                00020 ; VERSION 1.0      1979 STEPHEN GIBSON
                00030 ;
                00040 ;    RS-232 SIMULATION!!!!!
                00050 ;
                00060 ;FOR SIMPLEX TWO-WAY COMMUNICATION VIA
                00070 ;THE TRS-80 AUDIO CASSETTE PORT.
                00080 ;110 BAUD,HALF DUPLEX,NO LF,NO ECHO
                00090 ;
                00100 ;
7D00            00110        ORG    7D00H      ;STICK IT HERE
0002            00120 LOW    EQU    0002H      ;RS-232 0
0001            00130 HI     EQU    0001H      ;RS-232 1
007A            00140 T1     EQU    07AH       ;2 MS
0267            00150 LBD    EQU    267H       ;9 MS
0033            00160 CRT    EQU    33H        ;DISPLAY DRIVER
002B            00170 KBD    EQU    2BH        ;KEYBOARD DRIVER
0221            00180 PORT   EQU    221H       ;ROM PORT DRIVER
                00190 ;
                00200 ;
                00210 ;
7D00            00220 BEGIN  EQU    $          ;INITIALIZE
7D00 310080     00230        LD     SP,8000H   ;PUT STACK @ 7FFF
7D03 F3         00240        DI                ;STOP CLOCK
7D04 3E1C       00250        LD     A,1CH      ;HOME UP
7D06 CD3300     00260        CALL   CRT        ;WRITE
7D09 3E1F       00270        LD     A,1FH      ;CLEAR SCREEN
7D0B CD3300     00280        CALL   CRT        ;WRITE
7D0E CD167D     00290        CALL   RESET      ;INIT PORT
7D11 1816       00300        JR     RECV       ;ENTER MAIN LOOP
                00310 ;
                00320 ;
                00330 ;
7D13            00340 USR1   EQU    $          ;RESET PORT + 2MS DELAY
```

nor less than −17 dbm with the telephone coupled up to the modem and phone line dialed up to another party. Too much level is illegal, and too little won't let you make it into the first big time-share computer outside of town.

Depending on speaker efficiency or coupling, you can increase or decrease the send level by adjusting the primary taps on the transformer. A series resistor in the primary will also reduce the level. Don't get carried away with high speaker volume. A varistor in the phone will limit the absolute level on the line to around 0 dbm. It also generates odd harmonic distortion if overdriven. Keep your levels between −10 and −17 dbm, and you'll do fine.

Direct-connection users can

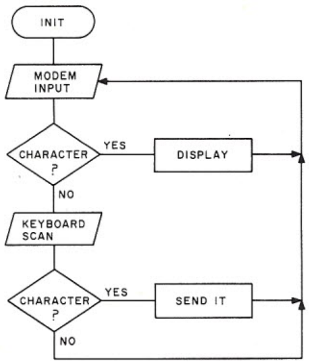


Fig. 7. A simple, duplex modem terminal-program flowchart.

tweak the output level by adjusting the value of R11 on the modem board. Adjust Rx for minimum send level at the opposite end of the transformer. You must, of course, make this final adjustment while connected to the line. Stick to the abovementioned send levels because the coupler arrangement has a varistor too.

Mickey Goes On-line

You may have a separate terminal or a terminal program running on your computer. If not, you can write a simple one from the flowchart in Fig. 7. The computer checks first with the modem port and outputs a character if present. Otherwise, it checks your terminal and/or loops around to check the modem again.

You can easily tie Mickey Modem to a serial port. Remember to investigate for RS-232-to-TTL compatibility. You can't mix 'em up without some hardware changes. If you don't have a spare serial port, you can tie Mickey to an unused bit on a parallel port and run the software UART program in Listing 1. The program is commented and should be easy for you to adapt to your machine.

TRS-80 users will find the program in Listing 2 useful. You simply tie Mickey Modem to your cassette port and remove the 100 to 220 Ohm resistor soldered to the cassette jack inside on the back of the main PC board. Of course, you can use the TRS-80 RS-232 board instead. Just be sure to observe the RS-232-to-TTL conventions mentioned previously.■

```
7D13 CDA37D   00350          CALL    DBOUN         ;DEBOUNCE 2 MS
7D16 210200   00360 RESET    LD      HL,LOW        ;RS232 MARK
7D19 CD2102   00370          CALL    PORT          ;SEND IT
7D1C C9       00380          RET                   ;DONE
              00390 ;
7D1D          00400 USR2     EQU     $             ;RESET PORT
7D1D CDA37D   00410          CALL    DBOUN         ;DEBOUNCE 2 MS
7D20 210200   00420          LD      HL,LOW        ;RS232 MARK
7D23 CD2102   00430          CALL    PORT          ;WRITE
7D26 C36D7D   00440          JP      KEYS          ;LOOP
              00450 ;
              00460 ;
              00470 ;
7D29          00480 RECV     EQU     $             ;SERIAL TO PARALLEL
7D29 010007   00490          LD      BC,700H       ;SET B TO 7, CLEAR C
7D2C 1600     00500          LD      D,00H         ;CLEAR D
7D2E DBFF     00510          IN      A,(0FFH)      ;GET BIT
7D30 17       00520          RLA                   ;SHIFT TO CY
7D31 303A     00530          JR      NC,KEYS       ;A SPACE BIT WILL SET CY
              00540 ; WE HAVE SOMETHING!
7D33 CD137D   00550          CALL    USR1          ;RESET PORT
7D36 DBFF     00560          IN      A,(0FFH)      ;GET BIT
7D38 17       00570          RLA                   ;MOVE TO CY
7D39 38E2     00580          JR      C,USR2        ;NOISE WOULD SET CY
              00590                                ;SO WE RESET AND EXIT
              00600 ;
              00610 ;WE NOW HAVE A VALID BYTE COMING IN AND WE
              00620 ;ARE IN THE MIDDLE OF THE SPACE BIT...SO
              00630 ;
7D3B CDA87D   00640 INPUT    CALL    DLY           ;DELAY 1 BIT TIME
7D3E DBFF     00650          IN      A,(0FFH)      ;GET BIT
7D40 E680     00660          AND     80H           ;MASK ALL BUT #7
7D42 2022     00670          JR      NZ,CHANGE     ;A TRANSITION!
7D44 7A       00680          LD      A,D           ;OTHERWISE,LOAD OLD BYTE
7D45 B1       00690 BACK     OR      C             ;GET PREVIOUS BITS
7D46 0F       00700          RRCA                  ;MOVE IN
7D47 4F       00710          LD      C,A           ;SAVE IN C
7D48 CD167D   00720          CALL    RESET         ;RESET PORT
7D4B 05       00730          DEC     B             ;COUNT BITS
7D4C 20ED     00740          JR      NZ,INPUT      ;DONE?
7D4E CDA87D   00750          CALL    DLY           ;IGNORE PARITY
7D51 CDA87D   00760          CALL    DLY           ;STOP BIT
7D54 CD167D   00770          CALL    RESET         ;RESET AFTER STOP BIT
7D57 79       00780          LD      A,C           ;GET BYTE
7D58 E67F     00790          AND     7FH           ;STRIP PARITY
7D5A FE60     00800          CP      60H           ;IS IT 60H OR MORE?
7D5C FA617D   00810          JP      M,TUBE        ;JP AROUND IF NOT
7D5F E65F     00820          AND     5FH           ;LOWER TO UPPER CASE
7D61 CD3300   00830 TUBE     CALL    CRT           ;WRITE
7D64 18C3     00840          JR      RECV          ;LOOK FOR MORE
7D66 A2       00850 CHANGE   AND     D             ;LAST BYTE
7D67 2F       00860          CPL                   ;INVERT BYTE
7D68 E680     00870          AND     80H           ;CLEAN UP
7D6A 57       00880          LD      D,A           ;SAVE
7D6B 18D8     00890          JR      BACK          ;SERIAL RE-ENTRY
              00900 ;
              00910 ;
              00920 ;
7D6D          00930 KEYS     EQU     $             ;READ KEYBOARD & SEND
7D6D CD2B00   00940          CALL    KBD           ;ANYONE
7D70 B7       00950          OR      A             ;THERE?
7D71 28B6     00960          JR      Z,RECV        ;SO LOOK AT PORT
7D73 E67F     00970          AND     7FH           ;CLEAR STUFF OFF
7D75 F5       00980          PUSH    AF            ;STORE COPY
7D76 CD3300   00990          CALL    CRT           ;WRITE YOURSELF A COPY
7D79 F1       01000          POP     AF            ;RETRIEVE
7D7A F680     01010 SEND     OR      80H           ;ADD 7TH BIT
7D7C F5       01020          PUSH    AF            ;SAVE BYTE
7D7D 210100   01030          LD      HL,HI         ;RS232 SPACE/START
7D80 CD2102   01040          CALL    PORT          ;WRITE
7D83 CDA87D   01050          CALL    DLY           ;4 1 BIT
7D86 0608     01060          LD      B,8H          ;8 BITS + STOP
7D88 37       01070          SCF                   ;+ STOP BIT
7D89 F1       01080 BITO     POP     AF            ;GET BYTE
7D8A 1F       01090          RRA                   ;PUT BIT IN CARRY
7D8B F5       01100          PUSH    AF            ;SAVE
7D8C 3005     01110          JR      NC,ZERO       ;CARRY=0
7D8E 210200   01120          LD      HL,LOW        ;CARRY=1,RS232 MARK
7D91 1803     01130          JR      OUT           ;OUTPUT
7D93 210100   01140 ZERO     LD      HL,HI         ;CARRY=0,RS232 SPACE
7D96 CD2102   01150 OUT      CALL    PORT          ;WRITE
7D99 CDA87D   01160          CALL    DLY           ;4 1 BIT
7D9C 05       01170          DEC     B             ;COUNT DOWN
7D9D 20EA     01180          JR      NZ,BITO       ;LOOP OUT BYTE
7D9F F1       01190          POP     AF            ;CLEAR STACK
7DA0 C3297D   01200          JP      RECV          ;BACK
              01210 ;
              01220 ;
              01230 ;
7DA3 217A00   01240 DBOUN    LD      HL,T1         ;2 MS
7DA6 1803     01250          JR      LOOP          ;DO IT
7DA8 216702   01260 DLY      LD      HL,LBD        ;110
7DAB 2B       01270 LOOP     DEC     HL            ;COUNT
7DAC 7C       01280          LD      A,H           ;MSB
7DAD B5       01290          OR      L             ;LSB
7DAE 20FB     01300          JR      NZ,LOOP       ;MORE?
7DB0 C9       01310          RET                   ;DONE
0000          01320          END
```

Jon Lindsay
3812 N. First St.
Fresno CA 93726

New Version of BASIC

Microsoft's BASIC-80 is designed for Z-80 and 8080 microprocessors.

From Microsoft comes an upgraded disk BASIC, BASIC-80, designed for the 8080 and Z-80 microprocessors. There are three versions: 8K, extended and disk. Though different system versions with minor variations are available, only the CP/M disk version of BASIC-80, called MBASIC, is reviewed here. For users of the old (version 4.51) MBASIC disk BASIC, most features of the new (version 5.03) MBASIC/BASIC-80 will be familiar (see Tables 1 and 2); but major features have been added! These improvements should please CP/M users who long for the subtleties found in a compiler such as CBASIC.

This review illuminates some less-used functions and commands of MBASIC/BASIC-80, and acquaints the reader with the expanded new features of the new CP/M version.

One interesting feature is the *protect option.* One possible reason that programmers use a compiler, especially in developing commercial programs, is because the resulting object code is difficult or impossible to modify. Hence, their program is not as subject to plagiarism. MBASIC/BASIC-80 has eliminated this one objection by including a new feature that protects the saved program source code from discovery. Once developed, the source program can be saved in a "protected" mode that only allows it to be run, not edited or listed. Not even executing the program with the old version of MBASIC will reveal its secrets.

This protected mode is entered by the command SAVE <program name>,P. This apparently places a direct statement, without line number, at the front of the program, thus faulting the list interpreter. Since it is also saved in an encoded binary format, its contents are further obscured. It cannot be "re-SAVEd" in an ASCII format and examined. Once in the protected mode, it appears to be there permanently. This requires that the original source listing or file be maintained in the unprotected mode in case any adjustments are necessary at a later date.

AUTO	LPRINT USING
CLEAR	LSET,RSET
CLOSE	MERGE
CONT	MID$
DATA	NAME
DEF FN	NEW
DEF INT/SNG/DBL/STR	NULL
DEF USR	ON ERROR GOTO
DELETE	ON...GOSUB
DIM	ON...GOTO
EDIT	OPEN
END	OUT
ERASE	POKE
ERR & ERL variables	PRINT
ERROR	PRINT USING
FIELD	PRINT#
GET	PRINT USING#
GOSUB...RETURN	PUT
GOTO	READ
IF...THEN...ELSE	REM
IF...GOTO	RENUM
INPUT	RESTORE
INPUT#	RESUME
KILL	RUN
LET	SAVE
LINE INPUT	STOP
LINE INPUT#	SWAP
LIST	TRON/TROFF
LLIST	WAIT
LOAD	WIDTH
LPRINT	

Table 1. Continued commands and statements.

New MBASIC/BASIC-80 Commands

The CALL statement calls an assembly-language subroutine at a certain address and passes a list of arguments to the subroutine.

The CHAIN statement allows interaction between the current program and another. The program can be "merged" with another, and all or part of the variables of the current program can be passed on to the new program. If only part of the variables are passed, the COMMON statement must be used to indicate which ones.

OPTION BASE declares the minimum value for array subscripts. The minimum value defaults to zero but can be assigned to one with the statement OPTION BASE 1.

RANDOMIZE (RND) reseeds the random number generator. If not used, the RND function will give the same random numbers sequence every time the program is executed. Interesting about its use is the automatic prompt asking for a "Random Number Seed (0-65529)" prior to executing the statement; it might improve the chance factor in your next game or simulation. The RND function no longer requires use of an argument (e.g., RND(34)). If none is supplied, it will assume a positive argument.

The WHILE <expression> . . . (loop statements) . . . WEND statement acts as a giant conditional statement. As long as the expression is not zero, the loop statements will continue to execute (see Example 1). When the program encounters WEND, it returns to WHILE, and the expression once again is checked for zero. If a zero expression is found, the program resumes with the next statement following WEND. Nesting of the WHILE/WEND statement may occur at any level; the programmer must remember that each WEND will match the most recent WHILE.

The last two new statements correct an inconvenience with the old MBASIC. The WRITE statement in new MBASIC outputs data at the terminal. Written as WRITE <list of expressions>, it is equivalent to a PRINT command, except that it prints commas between items in the expression list. The WRITE# command plays a more significant role by automatically overcoming the problem of writing two or more string expressions to a sequential disk file. Consider the following:

```
Let A$ = "Personal"
Let B$ = "Computer"
```

PRINT#1,A$;B$ writes "PersonalComputer" to the disk with no separation of the A$ and B$ string variables. If delimiters such as PRINT#1,A$;",";B$ are used, proper order is maintained and the separated variables may be detected by an INPUT# command. WRITE# handles the problem for you.

Some arithmetic manipulations that have been changed will affect programming. The conversion from floating-point to integer values results in rounding instead of truncation. Thus, C% = 4.5 results in C% = 5, where % represents an integer variable.

The body of a FOR . . . NEXT loop is skipped if the initial value of the loop times the sign of the step exceeds the final value times the sign of the step (see Example 2). The final value for the loop variable is always set before the initial value is set . . . the opposite of old MBASIC.

ABS	MID$
ASC	MKI$
ATN	MKS$
CDBL	MDS$
CHR$	OCT$
CINT	PEEK
COS	POS
CSNG	RIGHT$
CVI,CVS,CVD	RND
EOF	SGN
EXP	SIN
FIX	SPACE$
FRE	SPC
HEX$	SQR
INP	STR$
INSTR	STRING$
INT	TAB
LEFT$	TAN
LEN	USR
LOC	VAL
LOG	VARPTR
LPOS	

Table 2. Continued functions.

Division by zero and overflow no longer produce a program crash, as did old MBASIC. Instead, "division by zero" error message and machine infinity are printed, and the program continues to execute. However, if data from the errant calculation is important, an error trap may be necessary to save it or the logic of the program.

The rules for printing single-precision and double-precision numbers have been changed in new MBASIC. If the argument to ON . . . GOTO is out of range, an error message results and execution halts. Old MBASIC would probably have executed one of the arguments.

The CLEAR statement sets the highest memory location available for use by MBASIC/BASIC-80—needed to reserve space at the top of memory for assembly-language programs—and for setting aside stack space for BASIC. Unlike old MBASIC, which required declaration of string space beyond where a default occurred, the new BASIC allocates string space dynamically. If the CLEAR statement is omitted entirely, BASIC still finds room for that large word-processing file. An error occurs only when no free memory remains for MBASIC/BASIC-80 use.

A minor inconvenience is the new way of handling an INPUT statement. If a value has already been assigned to a variable, you cannot merely press RETURN to pass the assigned value to the same INPUT variable. It will not "fall through." A specific value must be given for the INPUT requested or an error message is printed. This more precise way of dealing with this command seems typical of the new philosophy regarding upgrading of this BASIC.

Two editing characters have been freed for special use with PRINT USING. The ampersand is used for variable length string fields, and the underscore demarks a character in a format string. Except as noted above, the ampersand is just another character that will no longer restart a current line.

The WIDTH statement, when written as Width 255, makes the line width infinite: no carriage return is issued at the end of a "normal line." This suggests an easy way of developing a self-managing text editor. The default width is still 72 characters.

BASIC will now recognize variable names up to 40 characters, as opposed to only two in old BASIC, and these may contain embedded reserved words. Because of this expanded capability, you cannot write a "compressed" BASIC program: ONDGOTO-100, 200,300ELSEPRINT"Not valid":GOTO-45, for example. Any program read from a disk in the compressed form produces an error. The solution is to expand the code so that a space exists between command and variable. A drawback is that you then lose the memory savings of the tighter code.

The INPUT$(X) function returns a string of X characters read from the terminal. Related in function to the INP function, INPUT$ allows entering data without echoing characters to the video display (see Example 3). It also conveniently handles string input without requiring a RETURN key. The argument X is the number of characters examined on input, not a port number assignment. The format INPUT$(X,#Y) allows data to be read from file #Y.

Oldies but Goodies

Several little-used but interesting functions are retained in new MBASIC, which, with minor exceptions, is compatible with the 4.51 version. Some of these functions are rarely seen in published programs, either because the casual programmer is unfamiliar with them or because he is attempting to make his program "universal." Some of these functions operate at the CPU level, so you can gain speed using them. Consider the following:

STRING$(n,m[X$]). This function returns a string of length n whose characters all have ASCII code m or the first character of X$. It prints a single character on the screen about twice as fast as a FOR . . . NEXT loop. To test this speed, compare the programs in Example 4.

VAL(X$). This function returns the numerical value of string X$. In Example 5, this allows use of consistent escape or interrogation commands, while permitting use of a convenient statement such as ON . . . GOTO. Thus, either a string *or* numeric response may be made.

INSTR ([I,] X$,Y$). Seeks the first occurrence of string Y$ in X$. An optional offset I sets the starting position of the search. Like STRING$, this function operates in CPU and returns a rapid reply in the form of a position number showing where Y$ begins within X$. This position number can be used as a switch to identify the presence or absence of a string. If one is present, the number returned will be greater than zero. The number can also be used with the TAB function to mark or underline a specific portion of the main string. Without this, you are obligated to use the much slower MID$ function.

INP(I). The first byte read from port I is returned; thus, information from/to a port may be detected. This can lead to an instantaneous response by the computer when a key is pushed; no RETURN is needed. Consider Example 6. A fairly standard terminal configuration, status port = 0; data port = 1, is used.

If all works correctly, pressing the E key will cause formation of a loop in which END PROGRAM is printed. Pressing any other key breaks the loop, and the program starts again. This technique might be valuable for people not familiar with computer operation. When a prompt says PRESS Y OR N

```
150 SWITCH=1: REM  SET TO "NOT-ZERO"
160 WHILE SWITCH
170       INPUT "Name ",N$
180       IF N$="QUIT" THEN SWITCH=0
190 WEND
200 PRINT:PRINT "Name list is done"
```

Example 1. WHILE/WEND statement.

```
10 X=0
20 FOR Y=1 TO X
30 PRINT Y
40 NEXT Y

      Old MBASIC          New MBASIC

         RUN                 RUN
          1                  Ok
         Ok
```

Example 2. The FOR . . . NEXT loop no longer "falls through."

```
10 PRINT "Welcome - what is your PASSWORD?"
20 P$=INPUT$(3)
30 IF P$="JON" THEN PRINT "You may proceed"
40 GOTO 20
```

Example 3. A use of INPUT$.

```
A.      10 C=1
        20 PRINT STRING$(70,ASC("-"))
        30 C=C+1: IF C=21 THEN STOP
        40 GOTO 20

B.      10 FOR X=1 TO 20
        20 FOR Y=1 TO 70:PRINT "-";:NEXT Y
        30 NEXT X
```

Example 4. Program A will execute at almost twice the speed of program B.

```
100 PRINT "Account numbers ---  1"
110 PRINT "Names -------------  2"
120 PRINT "Addresses ---------  3"
130 INPUT P$
140 IF P$="END" THEN 2000: REM  RETURN TO MAIN MENU
150 IF P$="?" THEN 3000: REM  GET HELP
160 ON VAL(P$) GOTO 500,700,900
170 GOTO 100
```

Example 5. Both string and numeric variables can be handled by use of the VAL statement.

and the person does, the computer becomes more interactive.

ERROR. Microsoft has continued with a liberal array of error traps. These become increasingly important as the program becomes more sophisticated. In Example 7, if the equation in line 660 has a zero for the denominator, line 650 has allowed for this by causing a branch to line 670, where the line number and type of error are identified. In this manner, the program will continue to operate and handle the predicted error any way the programmer chooses.

SWAP X,Y. This command exchanges values of two variables. As long as they are the same type, variables may be switched. SWAP X,Y assigns the value of X to Y and vice versa. It is handy in sorting routines.

DISK INPUT/OUTPUT. MBASIC/BASIC-80 supports both sequential and random files. Programs are stored on disk by using SAVE "filename" [,A]. Normally these programs are saved in a compressed binary format, which results in a savings in disk memory and increased speed in loading. In order to perform certain manipulations, however, it is sometimes necessary to save a program in ASCII format. Using the [,A] option after the filename easily accomplishes this.

To load a program from disk into memory, type LOAD "filename." Though present in memory, the program will not execute unless a RUN is typed or the program is loaded with the [,R] option after the filename. You can also use RUN "filename" to load and execute the program. Both commands may be placed within a program and executed from there. While data files can be kept open during chaining of programs, all variable values are reset to zero. To preserve these values, the CHAIN command must be used (see above).

To delete a file from the disk, type KILL "filename." To change the file name, type NAME "oldname" AS "newname." Both KILL and NAME can operate on all file types. If you desire the "protect" option, use the [,P] option (e.g., SAVE "filename",P).

Sequential files are supposed to be easier to create than random files, but once the basic procedure is understood, both sequential and random filing may be easily implemented. Sequential files are first OPENed and data PRINT#ed onto the disk by one of the following statements: PRINT#, PRINT # USING, WRITE#. When the operation is completed and further disk activity is not anticipated, the file must be CLOSEd. To read data into memory from disk, the file is again OPENed and read by the INPUT# or LINE INPUT# statements. Normally it is necessary to detect the end-of-file (EOF) to avoid attempting to read past the file, in which case an error results.

To increase disk I/O speed and manipulation of data, you must use random files, which use less memory because of the way the files are stored in disk (packed binary format), unlike sequential files (ASCII). You can directly access specific data without having to sequentially input and examine it. This is, perhaps, the greatest advantage of random file I/O.

A file is OPENed, as with sequential filing, but the additional information must be supplied to direct the computer to look toward the random buffer. The statement LSET (or RSET) is used for this purpose, and a PUT# statement deposits the data into the specified record. The file should be closed as soon as data transfer is completed. Reading data out from the disk into memory is the reverse of the process. The file must be opened and the computer FIELDed—space is allocated for the incoming data. A GET# statement plus record number moves the data into the designated memory location. This process involves more program code than a sequential file.

A new feature of random file I/O is the variable record size. Formerly a record size on a disk was 128 bytes, whether or not that many bytes had been set aside in the buffer during the data-saving process. It was possible to utilize "dummy" variables to allow writing in only part of the record space. While this increased the efficiency of data allocation, it did not change the 128-byte record length.

Now, the statement OPEN "R" ,1,"ACCT",21 opens file number 1 of a file named "ACCT" and restricts the record space on the disk to 21 bytes. Without this last number, the record size defaults to 128 bytes, the only record size in the old BASIC. This simple allocation saves disk space.

Summary

Microsoft's BASIC-80 is fast and efficient, and contains significant new features. In terms of memory, it costs the user 5858 bytes, the increased size over the former version. The expanded functions and enhancements on which programmers must rely to ease their work, however, seem worth the increased overhead. The documentation accompanying the language is clear and explicit. The various improvements suggest a more precise BASIC, more structured than before. Sixteen-digit double-precision math is allowed as well as sequential and random filing. Interaction between other programs via chaining should boost the use of modular programming, increasing the throughput of larger segmented programs. The "protect" mode should encourage commercial program development. The advantages of BASIC-80 should please *any* user. ■

```
10 WAIT 0,1,1
20 A=INP(1): REM  A = DECIMAL EQUIVALENT BYTE AT PORT 1
30 IF A>127 THEN A=A-128
40 IF A=ASC("E") THEN PRINT "END PROGRAM":GOTO 20
50 IF A=ASC("C") THEN PRINT "THIS PROGRAM WILL CONTINUE"
60 GOTO 10
```

Example 6. Spontaneous response is possible using INP.

```
650 ON ERROR GOTO 670
660 C1=((FNS(R1)-FNS(T1)*FNC(D))/(FNC(T1)*FNS(D)))
670 IF ERR=11 AND ERL=660 THEN RESUME 430
```

Example 7. Error trap for a division by zero. Note that even if no trap is used, the program will still execute; but any data from the equation will be lost. Trapping helps control data flow.

✓Reader Service—see page 258

Hashing Revisited

Turn your names into numbers and store and retrieve them with ease.

Raymond T. Vizzone
416 Stinson Ave.
Vacaville CA 95688

It had been four months since I last argued with my partner about the merits of hashing. In fact, I had introduced this technique to him, but was unable to convince him that it was the best way to store and retrieve data on disk or in arrays. He didn't like the disadvantages of rehashing—collision, wasted space and overhead. Neither did I! I couldn't blame him for ignoring me as he went back to his sorts and binary searches. I temporarily gave up.

While reading a book called *Compiler Construction for Digital Computers* by David Gries, I came across hashing again. At the same time, I also discovered some similarities existing between how Gries and Donald Fitchhorn ("DOCUFORM," *Kilobaud*, August 1978, p. 22) used hashing methods. With a little more research, I was ready to approach my partner with confidence.

Being subtle, I told my partner that hashing was the only way to retrieve data quickly. That was an absolute statement. He ignored me. I continued, nonetheless, and claimed hashing involves no sorting or binary or sequential searches. He had heard that song-and-dance before and was still waiting for more explanation. I emphasized that data is stored in memory sequentially as it is entered, never needing to be moved. Data is always available for quick retrieval with no need for rehashing. That did it! I had him now.

Implementing HASH-IT

In explaining the hashing method shared by both Gries and Fitchhorn, I used pictorials and illustrative examples. In the beginning, I proposed a function called HASH-IT, which would change names (KEYS) into numeric values called HASH-VALUES. The process is shown in Fig. 1. The function HASH-IT can be any method of converting an alphanumeric KEY to an integer numeric.

In using HASH-IT, each HASH-VALUE produced points into an array called the HASH TABLE (HT). For example, a HASH-VALUE of 3 is shown in Fig. 2. In turn, the contents of each element of the HASH TABLE point to an element of the STORAGE ARRAY (SA$), which holds the data and their respective KEYS (see Fig. 3).

I needed to implement two other concepts to fully explain the madness of this method. I wanted a pointer to show me where the next empty location of the STORAGE ARRAY was. For convenience, I called this pointer EMPTY. I made the STORAGE ARRAY two-dimensional to facilitate a linked-list affair. The use of this linked list will become clear in later examples. Fig. 4 shows a general pictorial of everything I needed to explain this hashing method.

To demonstrate this hashing method using Fig. 4, I used six names to "hash-in" to the STORAGE ARRAY. The first name was JOHNSON, which hashed to a numeric value of 3. The numbered location pointed to by EMPTY was then stored in the HASH TABLE at location 3. JOHNSON was stored in the STORAGE ARRAY at the location that EMPTY pointed to before being incremented (see Fig. 5). The names SMITH and JONES hashed to values of 6 and 1, respectively. Figs. 6 and 7 show how they were handled.

Rehash

The demonstration was going

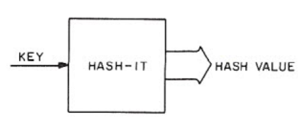


Fig. 1. HASH-IT function.

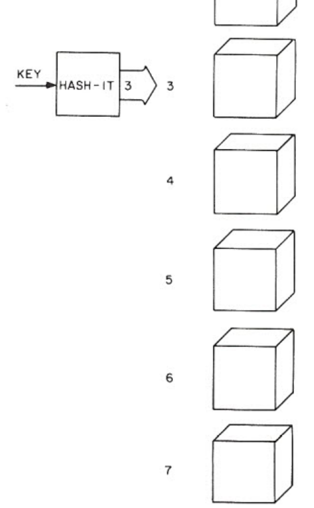


Fig. 2.

well, and my partner was still with me. All three names had hashed to different numeric values. So far, so good. I used the name MURPHY next and found it hashed to a value of 3. So had JOHNSON! This was a collision, and I had promised I wouldn't rehash or recalculate the HASH VALUE in order to find an empty element in the HASH TABLE. So now I had to use that second row of the STORAGE ARRAY.

MURPHY was stored at the next location in the STORAGE ARRAY pointed to by EMPTY. This location was 3, the sequential element after JONES. Normally, this value of 3 would be stored in the HASH TABLE at the location pointed to by the HASH VALUE. However, in this case, there was already a number there, more specifically, the pointer to JOHNSON. I couldn't disrupt this arrangement because I would not be able to find JOHNSON again! So I stored the value pointed to by EMPTY in the second row of the STORAGE ARRAY element containing JOHNSON.

To find MURPHY later would first require hashing it to a value of 3 (see Fig. 8). A look into the HASH TABLE at location 3 would find a pointer value of zero. The contents of the "zeroth" element of the STORAGE ARRAY would produce JOHNSON. JOHNSON would not be MURPHY by any means, so a further look would find that the link attached to JOHNSON pointed to the third element of the STORAGE ARRAY. Further investigation would find that MURPHY resided at this third element of the STORAGE ARRAY, and the search would be done.

The name DOE was next and gave me no trouble. It quietly hashed in and was stored as shown in Fig. 9.

JAMES was not so easy. It, too, hashed to a value of 3, right in there with JOHNSON and MURPHY. As handled before, the links were changed, and JAMES joined the group (see Fig. 10).

The Program

The program listed demonstrates the preceding pictorials and procedures. Its design is to provide a set of utility procedures to be used in other programs. The sections called HASH-IT and HASH-IN correspond to the figures above. I wrote them to simulate the ability to pass and receive parameters between procedures. Both procedures can be extracted and used in other programs.

The only change to make before HASH-IN can be truly considered general is to delete lines 520 through 524 and lines 527 and 529. These were written to facilitate the deletion ability of the program. HASH-IT is general and, as listed, will return a HASH VALUE (HASHV) for a given N$ not equal to a null string (""). The length of the HASH TABLE (LHT) must be given in

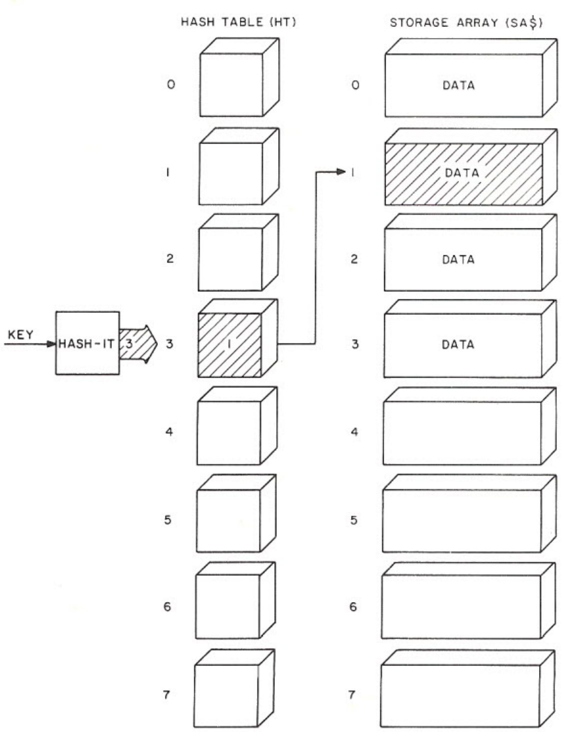


Fig. 3.

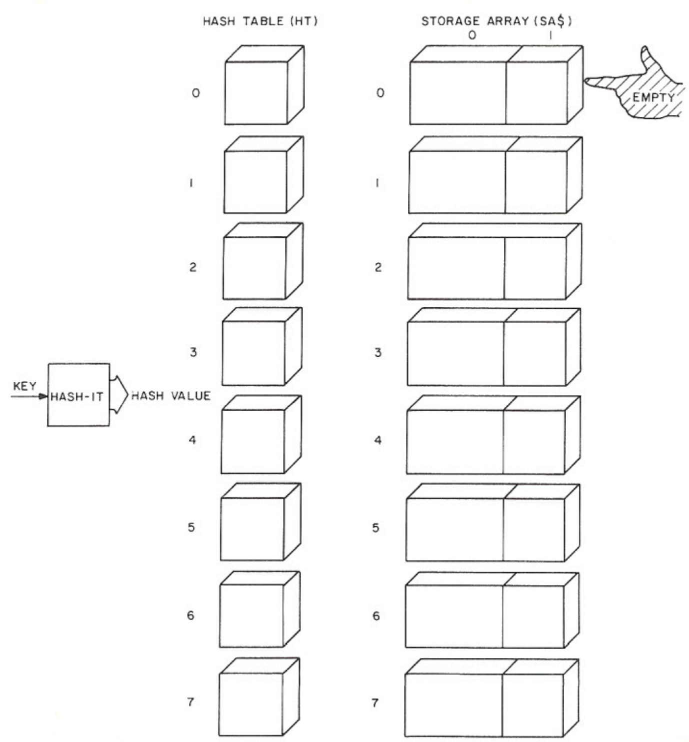


Fig. 4.

line 2120.

HASH-IT was developed by my partner, Mike Smith, to produce hash values from any length keys. His method first calculates the length of the key N$. Then, in lines 2050–2100, the ASCII values of every character in N$ are cumulatively added. Starting with the first character, every other character of N$ is added and stored in variable A(0). Simultaneously, every other character, starting with the second character of N$, is added and the sum stored in variable A(1).

This forms two separate numbers that are divided by 256, and their remainders are saved. The remainder from A(1) is multiplied by 256 and added to the remainder from A(0). The HASH VALUE is then made equal to this number divided by the variable LHT—the prime number size of the HASH TABLE. This all takes place in lines 2110–2120.

Uses

The program accepts and

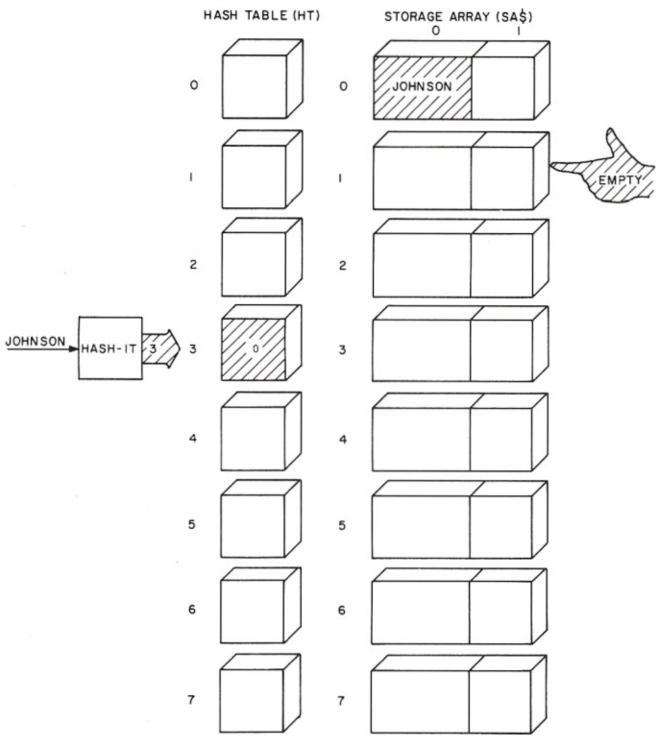


Fig. 5.

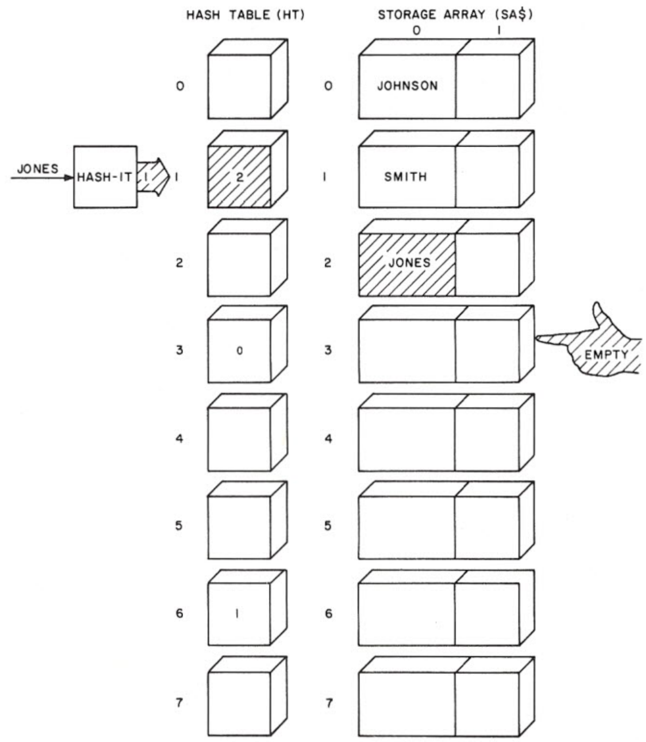


Fig. 7.

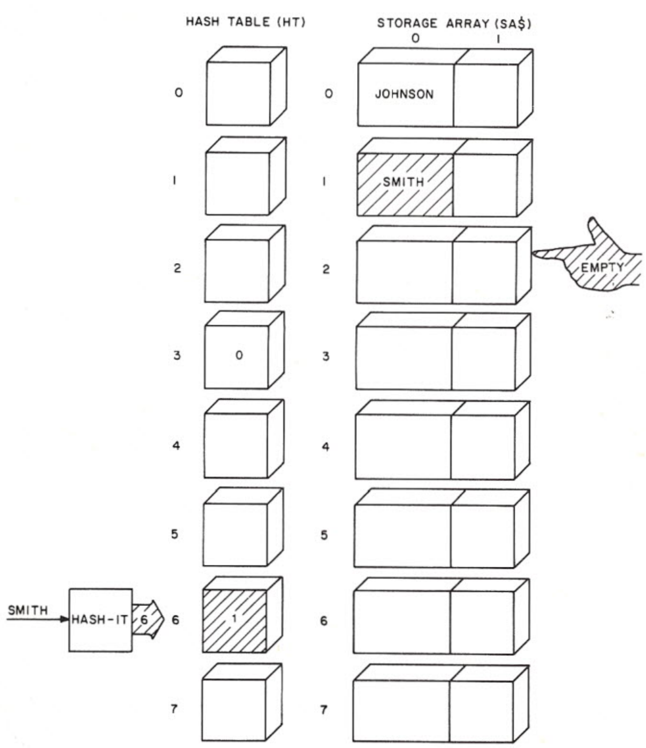


Fig. 6.

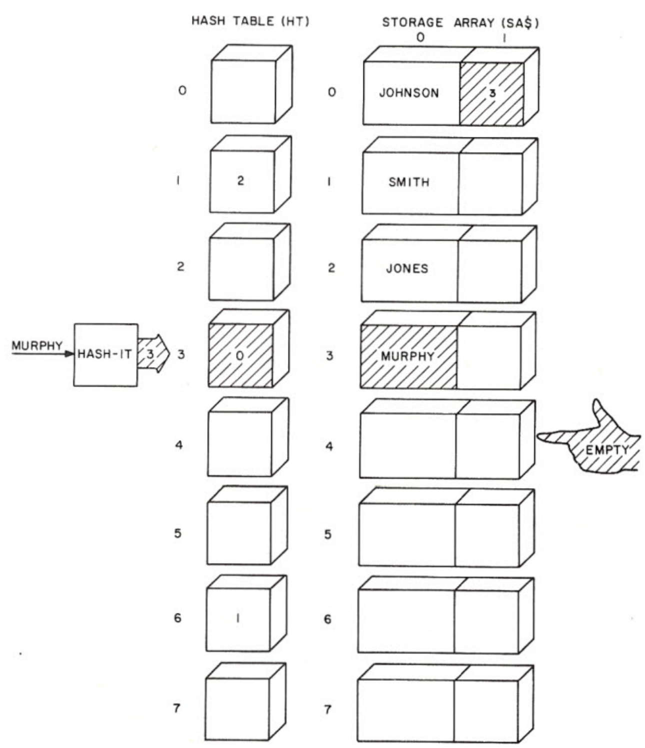


Fig. 8.

stores keys called SUBJECTS. Along with each SUBJECT, a 256-character miscellaneous field can be stored for later retrieval. One use of this program is to store names and phone numbers. The quick retrieval process used makes it a lot faster than your telephone book. Another idea is to provide an information network affair for clubs or others organizations. When members want to store subjects with information, they can use the ADD command. Then when members need to know further information about any subject, a query into the computer may find the desired information.

Quick retrieval and large storage capabilities make this sort of information mill a good application for a microcomputer. For a large data base, the addition of a data-save procedure can be added. A disk file can be substituted for the STORAGE ARRAY also.

Conclusion

This program can be used as is, or modified for more complex applications. The concept of hashing itself is in use with compilers and in DOCUFORM. Its explanation is intended to add another method of data storage and retrieval to a programmer's bag of tricks. ■

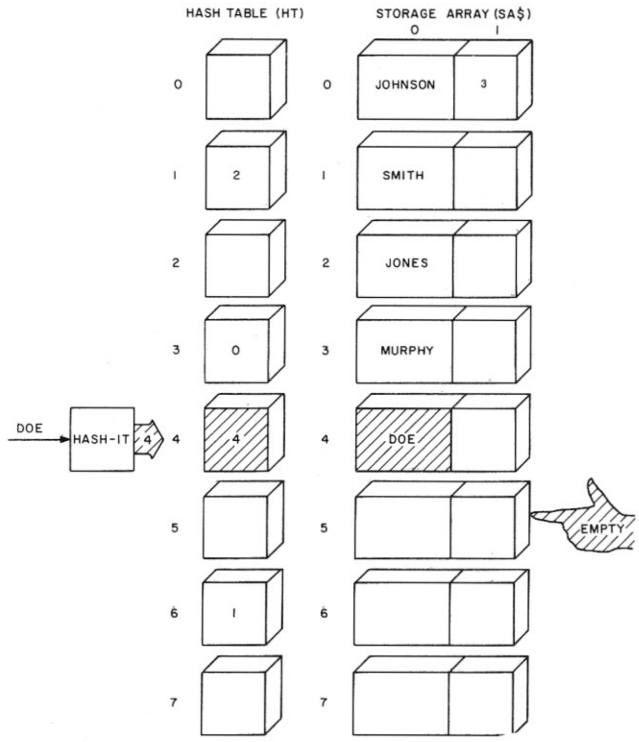


Fig. 9.

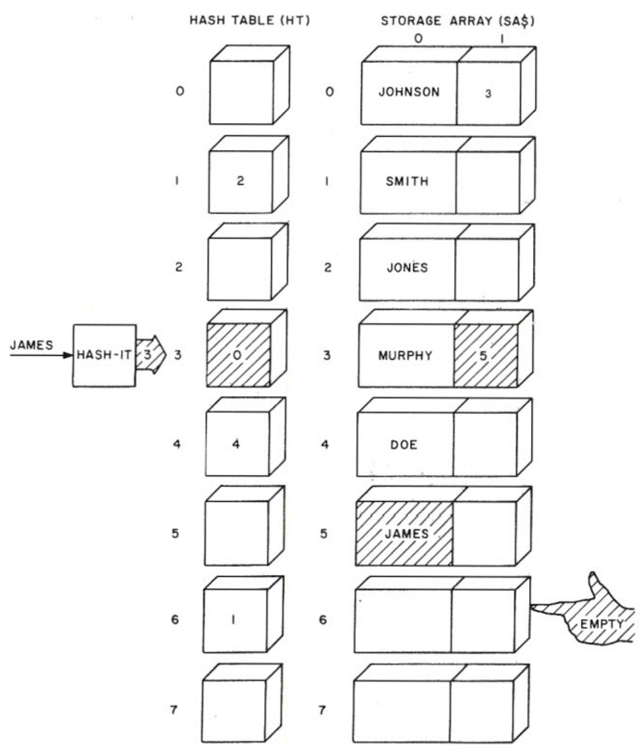


Fig. 10.

Program listing.

```
0 CLEAR : REM
                  INFORMATION MILL
1 REM ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
2 REM : WRITTEN BY R.T. VIZZONE     1979 CREATIVE CONSULTING.   :
3 REM : FOR THE APPLE II COMPUTER (REQUIRES APPLESOFT).          :
4 REM ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
5 REM
6 REM ::::::::::::::::::::::::
7 REM ::::  MAIN PROGRAM  ::::
8 REM ::::::::::::::::::::::::
9 REM
10  GOSUB 100: REM                  INTIALIZE
20  GOSUB 800: REM                  MENU
30  ON SEL GOSUB 300,900,1100,3000,1000,700
40  IF EXIT THEN 60
50  GOTO 20
60  END
70  REM
90  POKE 32,8: POKE 33,20: POKE 34,7: POKE 35,24: PRINT : RETURN
100  REM ::::::::::::::::::::::::
110  REM ::::  INITIALIZE  ::::
120  REM ::::::::::::::::::::::::
125 LHT = 101:LSA = 100: REM     DEFINE LENGHTS OF HASH TABLE AND STORAGE ARRAY
130  DIM HT(LHT),SA$(LSA,2)
140  FOR I = 0 TO LSA
150  ::SA$(I,0) = "-1"
160  ::SA$(I,1) = "-1"
170  NEXT I
180  FOR I = 0 TO LHT
190  ::HT(I) =  - 1
200  NEXT I
205 NLOC =  - 1:TR =  - 1
220  DEF  FN FOUND(A) = A + 1
230  RETURN
240  REM
300  REM ::::::::::::::::::::::::
310  REM :::::::  ADD  :::::::::
320  REM ::::::::::::::::::::::::
330  HOME
332  REM                         IS SA$ FILLED$
335  IF (TR < LSA) THEN 340: REM  NO IT'S NOT - CONTINUE.
336  REM                         YES IT IS! QUIT
337  VTAB (10): PRINT "NOT ENOUGH ROOM FOR AN ADD."
338  VTAB (12): PRINT "PRESS ANY KEY TO CONTINUE";: GET A$
339  GOTO 370
340  VTAB (5): INPUT "SUBJECT: ";N$
345  IF N$ = "" THEN 370: REM     IF USER QUIT THEN RETURN
347  VTAB (7): PRINT "INFO: "
348  PRINT
350  GOSUB 90: VTAB (7): HTAB (1): INPUT " ";MISC$: TEXT
355 ADD = 1: REM                   HASH-IN(N$,1,RN)
360  GOSUB 400
365  TEXT : HOME : VTAB (12): PRINT "THANK YOU"
367  FOR I = 1 TO 500: NEXT I: REM WAIT TO READ THANK YOU NOTE.
368 TR = TR + 1: REM               INCREMENT TOTAL RECORDS VARIABLE (TR)
370  RETURN
380  REM
400  REM ::::::::::::::::::::::::
410  REM : HASH-IN(N$,ADD,RN)  :
420  REM : GIVEN N$=NAME &     :
430  REM :       ADD=0 OR 1    :
440  REM : RETURN RN=RECORD #  :
450  REM ::::::::::::::::::::::::
455  REM
457  GOSUB 2000: REM                     HASH-IT(N$,HASV)
460 COLLISIEN = 0: REM                   COLLISION FLAG
470 P1 = HASHV: REM                      P1 NOW POINTS INTO THE HASH TABLE.
480 P2 = HT(P1): REM                     LOOK IN HASH TABLE
485  REM                                 ARE LOCATION CONTENTS EMPTY?
```

```
490  IF P2 < > - 1 THEN 600: REM         NO - GO CHECK SA$ POINTED TO BY P2
500  IF ADD THEN 520: REM                YES - DO AN ADD.
510 RN = - 1: GOTO 650: REM              YES BUT, THIS WAS A RETRIVE. RN=-1 FOR A RECORD NOT FOUND. RETURN.
515  REM                                 CHECK TO SEE IF THERE ARE ANY EMPTY LOCATIONS IN SA$ MADE BY A PREVIOUS DELETE.
520  IF  NOT DFLAG THEN 525
521  FOR I = 0 TO NLOC
522  IF SA$(I,0) = "-1" THEN 529: REM    THERE IS ONE - USE IT.
524  NEXT I
525 NLOC = NLOC + 1: REM                 THERE WASN'T ONE - MOVE THE EMPTY POINTER DOWN.
526 EMPTY = NLOC
527  GOTO 530
529 EMPTY = I:DFLAG = DFLAG - 1: REM     USING A PREVIOUSLY DELETED LOCATION. DECREMENT TOTAL # OF DELETED LOCATIONS.
530 P2 = EMPTY: REM                      AT LOCATION POINTED TO BY EMPTY...
540 SA$(P2,0) = N$: REM                  ...STORE KEY.
545 SA$(P2,2) = MISC$: REM               ATTACH ITS MISC INFO.
560  IF COLLISIEN THEN SA$(P1,1) =  STR$ (EMPTY): REM   THIS IS A COLLISION-ADD - CHANGE POINTERS.
570  IF  NOT COLLISIEN THEN HT(P1) = EMPTY: REM   THIS A NO COLLISION-ADD.
580  GOTO 650: REM                       ADD DONE - RETURN
590  REM                                 LOOK IN SA$ FOR A MATCH
600  IF SA$(P2,0) = N$ THEN 640: REM     A MATCH - RETURN.
610 P1 = P2:COLLISIEN = 1: REM           NO MATCH. SET COLLISION FLAG AND...
620 P2 =  VAL (SA$(P1,1)): REM             CHECK THE 2ND COLUMN OF SA$ FOR NEXT LOCATION TO SEARCH.
630  GOTO 485: REM                       SEARCH AGAIN.
640 RN = P2: REM                         RN NOW EQUALS THE LOCATION IN SA$ OF KEY JUST ADDED OR FOUND
650 ADD = 0
660  RETURN
700  REM ::::::::::::::::::::::
710  REM ::::::: EXIT :::::::
720  REM ::::::::::::::::::::::
730  REM
740 EXIT = 1
750  RETURN
800  REM ::::::::::::::::::::::
810  REM :::::: MENU :::::::
820  REM ::::::::::::::::::::::
830  HOME
840  VTAB (5): PRINT "SELECTION MENU"
850  VTAB (10): PRINT "1.  ADD A SUBJECT"
860  VTAB (12): PRINT "2.  FIND A SUBJECT"
870  VTAB (14): PRINT "3.  DELETE A SUBJECT"
875  VTAB (16): PRINT "4.  LIST ALL SUBJECTS"
879  VTAB (18): PRINT "5.  UPDATE A SUBJECT"
880  VTAB (20): PRINT "6.  EXIT PROGRAM": PRINT : PRINT
890  PRINT "INPUT SELECTION NUMBER PLEASE: ";
892  GET SEL$
894 SEL =  VAL (SEL$)
896  IF SEL < 1 OR SEL > 6 THEN 830
898  RETURN
900  REM ::::::::::::::::::::::
910  REM :::::: RETRIVE ::::::
920  REM ::::::::::::::::::::::
930  REM
940  HOME
950  VTAB (5): INPUT "SUBJECT: ";R$
955  IF R$ = "" THEN 999: REM            USER QUITS
960 ADD = 0:N$ = R$: GOSUB 400: REM      HASH - IN(R$,0,RN)
980  IF  FN FOUND(RN) THEN 990
982  VTAB (12): PRINT "I CAN'T FIND IT.  PRESS ANY"
983  PRINT "KEY TO GO ON. ";: GET A$
984  GOTO 999: REM                       RETURN
990  VTAB (7): PRINT "INFO: "
993  GOSUB 90: VTAB (7): HTAB (2): PRINT SA$(RN,2): TEXT
994  VTAB (24): PRINT "PRESS ANY KEY TO GO ON. ";:
995  GET A$
996  VTAB (24): HTAB (1): CALL  - 868
999  RETURN
1000  REM ::::::::::::::::::::
1010  REM :::: UPDATE :::::::
1020  REM ::::::::::::::::::::
1030  REM
1040  GOSUB 900: IF  NOT  FN FOUND(RN) OR R$ = "" THEN 1070: REM  IF USER QUIT OR NOT FOUND THEN RETURN
1041  VTAB (23): HTAB (1): CALL  - 868
1042  GOSUB 90
1047  VTAB (7): INPUT MISC$: REM         DISPLAY MISC INFO. UPDATE WITH CURSOR.
1060 SA$(FOUND,2) = MISC$
1070  TEXT : RETURN
1100  REM ::::::::::::::::::::
1110  REM ::::: DELETE ::::::
1120  REM ::::::::::::::::::::
1130  REM
1140  GOSUB 900: REM                     FIND RECORD REQUESTED
1145  IF R$ = "" THEN 1200: REM          IF USER QUIT OR...
1147  IF  NOT  FN FOUND(RN) THEN 1200: REM ...RECORD NOT FOUND THEN RETURN.
1150  VTAB (24): HTAB (1): PRINT "DO YOU WISH TO DELETE THIS INFORMATION ";: GET A$
1160  IF A$ = "N" THEN 1200
1170 SA$(RN,0) = "-1": REM               ERASE DELETED ENTRY.
1180  FOR I = 1 TO LHT: REM              CHECK DELETED ENTRY TO SEE IF...
1185  REM                                ...IT'S 1ST IN A SERIES OF COLLISIONS. IF SO...
1190  IF HT(I) < > RN THEN 1194: REM     ...REPLACE ITS POINTER IN HT WITH ITS COLLISION POINTER FROM ITS 2ND COLUMN
1192 HT(I) =  VAL (SA$(RN,1))
1193 SA$(RN,1) = "-1": GOTO 1195: REM    CONSEQUENTLY ERASE ITS 2ND COLUMN.
1194  NEXT I
1195 DFLAG = DFLAG + 1: REM              INCREMENT DFLAG TO INDICATE ANOTHER EMPTY LOCATION EXISTS.
1197 TR = TR - 1: IF TR < 0 THEN TR =  - 1: REM  DECREMENT TOTAL RECORDS, NOT TO BE <-1.
1200  RETURN
2000  REM ::::::::::::::::::::::
2010  REM : HASH-IT(N$,HASHV)  :
2020  REM : GIVEN N$, RETURN   :
2030  REM : HASHV              :
2040  REM ::::::::::::::::::::::
2042  REM                                DESCRIBED IN ACCOMPANYING ARTICLE
2045 HASHV = 0
2047 N =  LEN (N$)
2050 A(0) = 0:A(1) = 0
2060  FOR I = 1 TO N
2070 ::J =  INT ((I / 2 -  INT (I / 2)) * 2 + .05) *  SGN (I / 2)
2080 ::A =  ASC ( MID$ (N$,I))
2090 ::A(J) = A(J) + A
2100  NEXT I
2110 HASHV = ( INT ((A(0) / 256 -  INT (A(0) / 256)) * 256 + .05) *  SGN (A(0) / 256)) + 256 * ( INT ((A(1) /
      256 -  INT (A(1) / 256)) * 256 + .05) *  SGN (A(1) / 256))
2120 HASHV =  INT ((HASHV / LHT -  INT (HASHV / LHT)) * LHT + .05) *  SGN (HASHV / LHT)
2140  RETURN
3000  REM ::::::::::::::::::::
3010  REM :::: LIST ALL :::::
3020  REM ::::::::::::::::::::
3030  REM                                PRINT ALL ELEMENTS OF SA$ THAT ARE NOT EMPTY (NOT -1)
3035  HOME
3040  FOR I = 0 TO NLOC
3045  IF SA$(I,0) = "-1" THEN 3090
3050  HTAB (5): PRINT SA$(I,0)
3060 CV =  PEEK (37): REM                IS SCREEN FULL(I.E. IS CURSOR AT BOTTOM OF SCREEN)?
3070  IF CV < 21 THEN 3090: REM          NO - CONTINUE.
3075  PRINT : REM                        YES - CHECK WITH USER TO CONTINUE.
3080  PRINT "PRESS ANY KEY TO CONTINUE OR 'Q' TO QUIT";: GET A$: IF A$ = "Q" THEN 3110
3085  HOME
3090  NEXT I
3095  PRINT
3100  PRINT "PRESS ANY KEY TO GO ON ";: GET A$
3110  RETURN
```

TRS-80™

SYNCHRONOUS DATA SEPARATOR FOR DISK USERS.

SDS $34.50

THE SYNCHRONOUS DATA SEPARATOR WILL ELIMINATE 100% OF THE DISK READ ERRORS AND SPEED UP DISK ACCESS TIME BY ELIMINATING RETRIES BY THE DISK CONTROLLER.

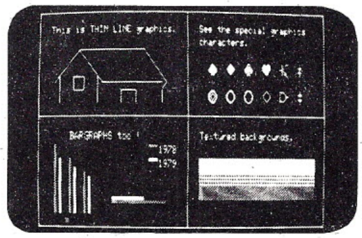
THE SDS PLUGS INTO THE DISK CONTROLLER'S SOCKET AND HAS ONLY TWO WIRES TO CONNECT. NO TRACES HAVE TO BE CUT. IT IS THE MOST RELIABLE AND ACCURATE DATA SEPARATOR AVAILABLE.

ADD $1.50 FOR SHIPPING AND HANDLING. CALIF. RESIDENTS ADD 6% SALE TAX. SEND CHECK OR MONEY ORDER TO:

G.P. ASSOCIATES
P.O. BOX 22822, SACRAMENTO, CA 95822

TRS-80 IS A TRADE MARK OF TANDY CORP. ✓83

LOWER CASE & GRAPHIC SYMBOLS GENERATOR KIT FOR TRS-80™ CG 16 ...... $94.50



TRUE 2 LINE DECENDER LOWER CASE, ELECTRONIC SYMBOLS, THIN LINE GRAPHICS, GAME SYMBOLS, TEXTURED BACKGROUNDS AND MANY MORE. DEMO CASSETTE IS INCLUDED.
FULLY ASSEMBLED WITH DETAIL INSTRUCTION FOR EASY INSTALLATION.
REQUIRES ELECT. PENCIL TYPE L.C. MOD OR ORDER MEMORY AND SWITCH KIT, SMK, FOR $18.50.

ADD $2.50 FOR S. & H., CALIF. RES. ADD 6% SALE TAX. SEND CHECK OR MONEY ORDER TO:

G.P. ASSOCIATES
P.O. BOX 22822, SACRAMENTO, CA 95822

Connect your TRS-80, Apple or ANY other computer to the phone lines.

USR-330 Originate—Auto-Answer Modem

- 0-300 Baud
- Stand Alone
- RS232
- 1 Year Warranty
- Crystal Controlled
- Bell 103/113

$339

- State of the Art LSI circuitry
- 5 stage active filters

FCC certified for direct connection to phone lines via standard extension phone jack

Call or write for free literature

U.S. ROBOTICS, INC.
1035 W. LAKE ST.
CHICAGO, ILL. 60607 ✓181
(312) 733-0497

C-10 **SHORT CASSETTES** 50 FT.

Qty.	Price
10	$0.75
50	$0.65

Premium tape and cassettes acclaimed by thousands of repeat order microcomputer users. Price includes labels, cassette box and shipping in U.S.A. VISA and M/C orders accepted. California residents add sales tax. Phone (415) 968-1604. ✓123

MICROSETTE CO.
475 Ellis Street
Mt. View, CA 94043

PROGRAMS for OSI 65U V1.2 1-22-80 or later update including:

1) OSI newsletter info incorporated into programs
2) enhancements for operators
3) useful utility and demos
4) rewrite of 'disk' manual

Send this adv. copy and a) original OSI 8" disk with $10.00; or b) $15.00 for disk without system.
(Indiana residents add 4% sales tax)

TERMS: The buyer agrees not to sell or give the disk or its info. to others.

Name ____________________

Address ____________________

City State Zip ____________________

C & J Supply ✓258
Box 806, Marion, IN 46952

alphanetics ✓265

TRS-80 © TAPE DIGITIZER

ALSO AVAILABLE WITHOUT CASSETTE REMOTE ON/OFF SWITCH FOR

$54.95

Used by the U.S. Coast Guard and U.S. Navy

- ELIMINATES CASSETTE LOADING AND COPYING PROBLEMS . . . EVEN "SYSTEM" TAPES!
- MAKES TAPE PROGRAM LOADING PRACTICALLY INDEPENDENT OF VOLUME CONTROL SETTING!
- MAKES PERFECT COPIES OF ANY TAPE DIGITALLY WITHOUT USING COMPUTER, DIGITIZED TO EXACT REPLICA OF TR-80's SIGNAL WHILE REMOVING HUM, NOISE AND OTHER MINOR DROPOUTS
- A.C. POWERED NO BATTERIES CASSETTE SWITCH ALLOWS MANUAL OR COMPUTER CONTROL OF CASSETTE RECORDER
- FEED YOUR CASSETTE TO THE TAPE DIGITIZER AND FEED YOUR COMPUTER THE EXACT DIGITAL WAVEFORM THE TRS-80 GAVE TO THE TAPE WHILE MAKING A COPY AT THE SAME TIME!
- THE TAPE DIGITIZER IS COMPLETELY COMPATIBLE WITH LEVEL I AND II

"NEW IMPROVED!"

- *"GOOD DATA" INDICATOR EASILY ENABLES SETTING VOLUME CONTROL FOR GOOD PROGRAM AND DATA LOADS EVERY TIME!*
- *GET RID OF YOUR TAPE BUGS TODAY FOR ONLY $59.95 POSTAGE PAID GUARANTEED TO FIX YOUR TAPE PROBLEMS OR RETURN IN 10-DAYS FOR A FULL REFUND!*

24-hour phone
(707) 887-7237

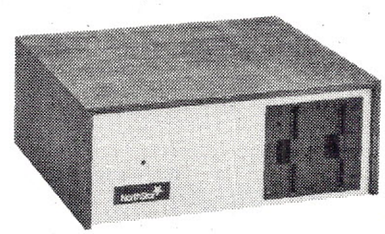
ACU-DATA ALPHANETICS P.O. BOX 597, FORESTVILLE, CALIFORNIA 95436

Save more than 20%!

AMERICAN SQUARE COMPUTERS

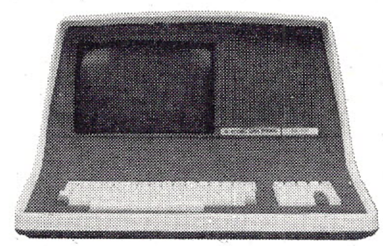
NORTH STAR--INTERTUBE THINKER TOYS--MICROTEK

The smartest computers at the smartest price.

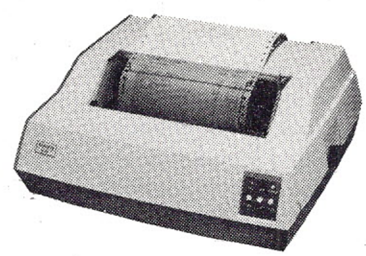


HORIZON QUAD & DOUBLE DENSITY

	LIST	ONLY
HORIZON-1-32K-D KIT	$1999	$1585
HORIZON-2-32K-D KIT	$2399	$1905
ASSEMBLED & TESTED	$2765	$2195
HORIZON-2-32K KIT QUAD	$2799	$2225
ASSEMBLED & TESTED	$3215	$2555
PASCAL FOR NORTH STAR ON DISK		49
POWERFUL NORTH STAR BASIC		FREE
TEI PT 212 COMPUTER 5 MHz	8000	6250
THINKER TOYS DISCUS/2D A&T	1149	949
DISCUS/2+2 1.2 MEGABYTES A&T	1549	1299
MEASUREMENT SYSTEM MEMORY A&T 4 MHz 64K		640
GODBOUT MEMORY		CALL FOR PRICE



INTERTUBE II SMART TERMINAL	$995	$745



MICROTEK PRINTER	750	675
ANADEX PRINTER	995	875
FLORIDA DATA PRINTER 600 CPS	4300	CALL FOR PRICE

Super Software at Reasonable Prices

MARYELLEN WORD PROCESSOR YOUR BEST BUY	$38
TEXTWRITER III	$125
EZ-80 TUTORIAL—LEARN MACHINE LANGUAGE	$25
PDS FOR NORTH STAR—BETTER THAN CP/M	$99
COMPILER FOR HORIZON—SECRET SUPERFAST CODE	$100

10% Off Software Prices with Computers

VERBATIM THE BEST DISKETTES BOX OF 10	$29
WHICH COMPUTERS ARE BEST? BROCHURE	FREE
NORTH STAR DOCUMENTATION, REFUNDABLE WITH COMPUTER	$20

AMERICAN SQUARE COMPUTERS ✓56

KIVETT DR
JAMESTOWN NC
27282
(919) 883-1105

6800 Trace and Disassemble Program

This program puts you on the trail of runaway routines.

Richard Carickhoff
812 Pulaski Dr.
Lansdale PA 19446

Did you ever write a program that didn't work and then spend hours, or even days, debugging it? Did you ever wonder how the program got to that particular location? . . . why that compare instruction wasn't working as you thought it should? . . . why that multiply routine didn't work?

Well, I've been down that road many times myself, so I decided to write a program that would allow me to trace a program instruction by instruction while, at the same time, see exactly what was taking place before and after the execution of each instruction.

The 6800 Trace and Disassemble program does just that. The program enables the user to perform the following functions:

- Program trace function
- Go to user's program function
- Program disassemble function
- Memory examine and change function
- Register examine and change function

The detailed explanations, along with operating procedures for each of these functions, are described in the following paragraphs.

At the start of each function it is assumed that the last data character printed by the terminal is a colon (:), which is the program's prompt character. All values entered must be in hexadecimal format.

Program Trace Function

The program trace function will trace the user's program one instruction at a time, while outputting to the terminal the location, mnemonic, operand, contents of all MPU registers (CC, B, A, X, SP) and the next return address in the stack. The trace function will do this for each instruction prior to its execution.

The trace funtion can be performed by typing one of the following two responses:

: T nnnn
or
: T nnnn, mmmm

The first response must be terminated with a carriage return. The character T specifies a trace function. The four hexadecimal digits following T specify the starting address of the first instruction to be traced. This response instructs the program to trace only one instruction (see Example 1).

At this point the trace function waits for the operator to enter a character. If the character is any character other than the Escape (1B hex), the instruction displayed will be executed and the next instruction will be output to the terminal along with the contents of all the MPU registers (see Example 2).

The contents of the following MPU registers are printed along with each instruction:

cc—Condition code register
b—B register
a—A register
x—X register
sp—Stack pointer

```
:T 0103                  cc   b    a    x     sp    rtn
0103   JMP    0225       8F   19   FF   2242  A049  7B05
```

Example 1.

```
:T 0103                  cc   b    a    x     sp    rtn
0103   JMP    0225       8F   19   FF   2242  A049  7B05
0225   LDS    22         CF   19   FF   2242  A049  7B05
```

Example 2.

rtn—First return address at the top of the stack

The contents of the program counter is the location of the instruction to be executed.

With the use of the trace function, the operator can step through his program one instruction at a time. The contents of all the MPU registers are always visible before and after the execution of each instruction. Also, the instruction is always printed before it is executed so the operator can decide whether to terminate the trace at that point (depressing Escape key) or to continue.

The second response to the prompt character is used to trace a program until the breakpoint address is reached. The first four hexadecimal digits define the starting address of the first instruction of the trace sequence. The second four hexadecimal digits following the comma define the breakpoint address. Once the last digit is entered, the program will immediately start tracing the program starting at the start address.

The output format is the same as the single trace function except that the program will continue outputting each instruction until the breakpoint address is reached. At that point the trace function operates in the same manner as the single trace function. That is, depressing the Escape key terminates the trace and depressing any other key executes the last instruction printed and outputs the next instruction. The Escape key is also used to terminate a trace sequence prior to reaching the breakpoint address.

Caution: The trace function traces a program with the use of the software interrupt (SWI). Always terminate any trace sequence using the Escape key. Using the system reset may leave a software interrupt in the user's program.

This method of tracing a program is normally used to determine how a program arrived at a particular location. If a CRT is being used for a terminal, the last 15 instructions executed will still appear on the screen (assuming the CRT has a minimum of 16 lines). The rate at which the program executes is controlled by the output rate of the terminal being used.

Program A shows an example of the trace function. The program selected is Tom Pittman's 6800 Tiny BASIC. I chose this program because it is well known and is an interesting program to trace. It also demonstrates the visibility of a program using the trace function.

The starting address was set at 0103 hex, which is Tiny BASIC's warm start address. The breakpoint address was set at an address that would not be reached. This allowed me to terminate the program at any point during the trace.

In Program A there are several instructions that are disassembled with asterisks (***) for the mnemonic and ROM for the operand. This alerts the operator that the trace function came upon a ROM address that could not be loaded with the software interrupt. The trace function in this case places the software interrupt at the return address. The trace function assumes that routines in ROM are functional and always return via the RTS (return subroutine) instruction.

The ROM address shown in Program A is the MIKBUG output routine (EIDI). Examining the contents of the A register prior to executing the output routine shows the character being output. Also, the output is reflected in the trace printout as indicated by the line feed following the first output by Tiny BASIC.

```
:T Ø1Ø3,ØFFF
Ø1Ø3  JMP    Ø225    C1 19 ØD 2242 AØ7D Ø22A
Ø225  LDS    22      C1 19 ØD 2242 AØ7D Ø22A
Ø227  JSR    Ø62C    C9 19 ØD 2242 AØ7F ØØØØ
Ø62C  LDA A #ØD      C9 19 ØD 2242 AØ7D Ø22A
Ø62E  BSR    Ø649    C1 19 ØD 2242 AØ7D Ø22A
Ø649  CLR    ØØBF    C1 19 ØD 2242 AØ7B Ø63Ø
Ø64C  JMP    Ø598    C4 19 ØD 2242 AØ7B Ø63Ø
Ø598  INC    ØØBF    C4 19 ØD 2242 AØ7B Ø63Ø
Ø59B  BMI    Ø5A7    CØ 19 ØD 2242 AØ7B Ø63Ø
Ø59D  STX    BA      CØ 19 ØD 2242 AØ7B Ø63Ø
Ø59F  PSH B          CØ 19 ØD 2242 AØ7B Ø63Ø
Ø5AØ  JSR    Ø1Ø9    CØ 19 ØD 2242 AØ7A 19Ø6
Ø1Ø9  JMP    E1D1    CØ 19 ØD 2242 AØ78 Ø5A3
E1D1  ***    ROM

Ø5A3  PUL B          CØ 19 ØD 2242 AØ7A 19Ø6
Ø5A4  LDX    BA      CØ 19 ØD 2242 AØ7B Ø63Ø
Ø5A6  RTS            CØ 19 ØD 2242 AØ7B Ø63Ø
Ø63Ø  LDA B Ø111     CØ 19 ØD 2242 AØ7D Ø22A
Ø633  ASL B          CØ Ø3 ØD 2242 AØ7D Ø22A
Ø634  BEQ    Ø63E    CØ Ø6 ØD 2242 AØ7D Ø22A
Ø636  PSH B          CØ Ø6 ØD 2242 AØ7D Ø22A
Ø637  BSR    Ø642    CØ Ø6 ØD 2242 AØ7C Ø6Ø2
Ø642  CLR A          CØ Ø6 ØD 2242 AØ7A Ø639
Ø643  TST    Ø111    C4 Ø6 ØØ 2242 AØ7A Ø639
Ø646  BPL    Ø649    CØ Ø6 ØØ 2242 AØ7A Ø639
Ø649  CLR    ØØBF    CØ Ø6 ØØ 2242 AØ7A Ø639
Ø64C  JMP    Ø598    C4 Ø6 ØØ 2242 AØ7A Ø639
Ø598  INC    ØØBF    C4 Ø6 ØØ 2242 AØ7A Ø639
Ø59B  BMI    Ø5A7    CØ Ø6 ØØ 2242 AØ7A Ø639
Ø59D  STX    BA      CØ Ø6 ØØ 2242 AØ7A Ø639
Ø59F  PSH B          CØ Ø6 ØØ 2242 AØ7A Ø639
Ø5AØ  JSR    Ø1Ø9    CØ Ø6 ØØ 2242 AØ79 Ø6Ø6
Ø1Ø9  JMP    E1D1    CØ Ø6 ØØ 2242 AØ77 Ø5A3
E1D1  ***    ROM
Ø5A3  PUL B          C1 Ø6 ØØ 2242 AØ79 Ø6Ø6
Ø5A4  LDX    BA      C1 Ø6 ØØ 2242 AØ7A Ø639
Ø5A6  RTS            C1 Ø6 ØØ 2242 AØ7A Ø639
Ø639  PUL B          C1 Ø6 ØØ 2242 AØ7C Ø6Ø2
Ø63A  DEC B          C1 Ø6 ØØ 2242 AØ7D Ø22A
Ø63B  DEC B          C1 Ø5 ØØ 2242 AØ7D Ø22A
Ø63C  BNE    Ø636    C1 Ø4 ØØ 2242 AØ7D Ø22A
Ø636  PSH B          C1 Ø4 ØØ 2242 AØ7D Ø22A
Ø637  BSR    Ø642    C1 Ø4 ØØ 2242 AØ7C Ø4Ø2
Ø642  CLR A          C1 Ø4 ØØ 2242 AØ7A Ø639
Ø643  TST    Ø111    C4 Ø4 ØØ 2242 AØ7A Ø639
Ø646  BPL    Ø649    CØ Ø4 ØØ 2242 AØ7A Ø639
Ø649  CLR    ØØBF    CØ Ø4 ØØ 2242 AØ7A Ø639
Ø64C  JMP    Ø598    C4 Ø4 ØØ 2242 AØ7A Ø639
Ø598  INC    ØØBF    C4 Ø4 ØØ 2242 AØ7A Ø639
Ø59B  BMI    Ø5A7    CØ Ø4 ØØ 2242 AØ7A Ø639
Ø59D  STX    BA      CØ Ø4 ØØ 2242 AØ7A Ø639
Ø59F  PSH B          CØ Ø4 ØØ 2242 AØ7A Ø639
Ø5AØ  JSR    Ø1Ø9    CØ Ø4 ØØ 2242 AØ79 Ø4Ø6
Ø1Ø9  JMP    E1D1    CØ Ø4 ØØ 2242 AØ77 Ø5A3
E1D1  ***    ROM
```

Program A.

Trace Function Restrictions

There are only two restrictions on the trace function. The first is that it will not trace a program that uses a software interrupt, since the software interrupt interferes with the trace function's software interrupt. The second restriction is that the trace function cannot be used to trace itself.

Go to User's Program Function

This function allows the operator to execute his program. The operator may specify a breakpoint address in order to return to the trace program. This function can be performed by typing one of the following two responses:

: G nnnn

or

: G nnnn, mmmm

The first response must be terminated with a carriage return. The character G specifies a Go function. The four hexadecimal digits following G specify the starting address of the program to be executed (e.g., : G 0103).

The only way to return to the Trace and Disassemble program with this response is through the system monitor.

The second response is used to execute a user's program until the breakpoint address is reached. The first four hexadecimal digits define the starting address of the program to be executed. The second four hexadecimal digits following the comma define the breakpoint address. Once the last digit is entered, the MPU will start executing the user's program. Once the breakpoint address is reached, the control of the program is returned to the trace function (see Example 3).

The program can be traced from this point one instruction at a time by simply depressing any key other than the Escape key. The trace will operate in the same manner as if a trace function was being performed.

```
: G 0103,022F
022F   LDX   #0080   C1   00   00   07A1   0000
```

Example 3.

If the program does not reach the breakpoint address and the operator wishes to return to the trace and disassemble program, he must perform a system reset and return through the system monitor. However, the software interrupt still exists at the breakpoint address.

To remove the interrupt and replace it with the original instruction, the Go to User's Program function can be executed where the starting address is set to the breakpoint address. The program will immediately return, displaying the original instruction at the terminal. The operator can then terminate the trace function by depressing the Escape key.

```
: D 0225
0225   9E   22   LDS   22
```

Example 4.

```
: D 0225
0225   9E   22     LDS   22
0227   BD   062C   JSR   062C
```

Example 5.

Program Disassemble Function

This function allows the operator to disassemble any 6800 program including the Trace and Disassemble program itself. The disassemble function can disassemble one instruction at a time or a sequence of instructions, while outputting to the terminal the location, object code, mnemonic and operand for each instruction.

The disassemble function can be performed by typing one of the following two responses:

: D nnnn
or
: D nnnn, mmmm

The first response must be terminated with a carriage return. The character D specifies a disassemble function. The four hexadecimal digits following D specify the starting address of the instruction to be disassembled (see Example 4).

At this point the disassemble function waits for the operator to enter a character. If the character is any character other than an Escape, the next instruction in sequence will be disassembled (see Example 5). In doing so, the operator can step through a disassembly of a program one instruction at a time.

The second response is used to disassemble a list of instructions. The first four hexadecimal digits specify the first instruction to be disassembled.

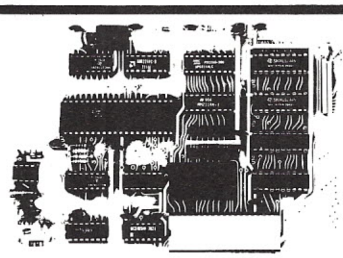
```
:D 0225,0280
0225 9E 22     LDS     22
0227 BD 062C   JSR     062C
022A FE 01FE   LDX     01FE
022D DF 2A     STX     2A
022F CE 0080   LDX     #0080
0232 DF C2     STX     C2
0234 CE 0030   LDX     #0030
0237 DF C0     STX     C0
0239 9F 26     STS     26
023B 8D B8     BSR     01F5
023D 8D 07     BSR     0246
023F 20 FA     BRA     023B
0241 8C 1066   CPX     #1066
0244 20 F3     BRA     0239
0246 CE 0117   LDX     #0117
0249 DF BC     STX     BC
024B 81 30     CMP A   #30
024D 24 56     BCC     02A5
024F 81 08     CMP A   #08
0251 25 91     BCS     01E4
0253 48        ASL A
0254 97 BD     STA A   BD
0256 DE BC     LDX     BC
0258 EE 17     LDX     17,X
025A 6E 00     JMP     00,X
025C BD 062C   JSR     062C
025F 86 21     LDA A   #21
0261 97 C1     STA A   C1
0263 BD 0109   JSR     0109
0266 86 80     LDA A   #80
0268 97 C3     STA A   C3
026A D6 2B     LDA B   2B
026C 96 2A     LDA A   2A
026E F0 01FF   SUB B   01FF
0271 B2 01FE   SBC A   01FE
0274 BD 0542   JSR     0542
0277 96 C0     LDA A   C0
0279 27 0F     BEQ     028A
027B CE 0293   LDX     #0293
027E DF 2A     STX     2A
0280 BD 05AD   JSR     05AD
```

Program B. Disassemble function.

The second four hexadecimal digits following the comma specify the last instruction to be disassembled.

Once the last digit is entered, the program will immediately list each instruction in sequence until the last address is reached. The last address specified must be on an instruction boundary. Otherwise, the disassembly will continue past the last address. The Escape key can be used to terminate any list sequence.

When the last address is reached, the disassembly will stop. The operator can continue the disassembly one instruction at a time by depressing any key other than Escape. Otherwise, the Escape key will terminate the disassembly and return control back to the control



New **Communications/Control CPU Card**

- **CPU-Z-80 1K RAM**
  16 bit interval timer and interrupt
  EPROM* 2708 standard (2716 optional)
- **Serial Communications**-RS-232 interface, UART Complete MODEM capability, programmable baud rates, etc.
- **Parallel I/O**-16 bits in (TTL), 16 bits Out (TTL)
- **Power**-on and external reset

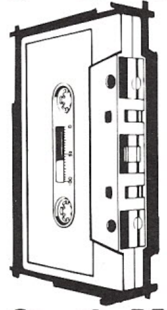
*EPROM not included

$195 assembled, tested, with warranty and documentation.

Immediate delivery from stock!
All orders shipped prepaid ✓135
MC, VISA, phone orders welcome
Utah residents add 4¾% sales tax

Vantage Data Products
550 West 200 South
Suite 8
Provo, Utah 84601
(801) 377-6687

FOR MINI-COMPUTERS
FREE Special Trial Offer!!



When you order 23 C-10 cassettes featuring Precision transport mechanism and Digital pressure pad. (In individual, dust-proof, plastic boxes with labels) for a low introductory price of $19.95, we will send you FREE 1 C-10 cassette for you to test. If you are not delighted with its performance you can return the unused 23 units for a full refund. Please add $1.05 to cover postage and handling. N.Y. Res. add 7%. VISA and Master Charge accepted.

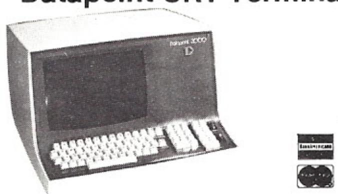
Write for pricing on larger quantities.

Studio Magnetics Co., Inc.
✓179

SM

83 Carlough Road
Bohemia, N.Y. 11716
516-589-4300

Datapoint CRT Terminals



Fully-Assembled — Guaranteed

#3000 $495 **#3360 $545**

- Add $15 packing **refurbished**
- Shipped FOB Washington, D.C. Terms check, M.O. or charge.
- 90-day guarantee • **Scrolling version 3360 $595**

Model 3360 speeds from 300-4800 Baud, numeric keypad, cursor controls, Edit, Block-Transmit, search modes, ASCII Keyboard with codeable options. Green phosphor, 24 80 ch lines, addressable cursor; RS-232C serial interface; other speeds available. Manual $10; cable kit $9.95. Datashare version 3360 $595. Model 3000 $675. M-33 KSR Teletype $395. QUANTITY DISCOUNTS AVAILABLE. Leasing, service at low prices.

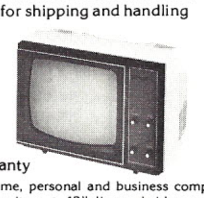
TELECOMMUNICATIONS SERVICES CO.
Box 4117, Alexandria, Va. 22303
703-683-4019 / TLX 89-623 ✓131

12" BLACK & WHITE LOW COST VIDEO TERMINAL

Easily interfaced with Radio Shack TRS 80

$159.00 LIST

Will sell 6 feet coaxial cable **$5.00**
Texas residents add 5% sales tax.
Add $5 for shipping and handling



One year limited warranty

• Ideal for home, personal and business computer systems; surveillance monitors • 12" diagonal video monitor • Composite video input • Compatible with many computer systems • Solid-state circuitry for a stable & sharp picture • Video bandwidth—12 MHz ± 3 DB • Input impedance—75 Ohms • Resolution—650 lines Minimum IN Central 80% of CRT; 550 Lines Minimum beyond central 80% of CRT ref EIA RS-375 • Dimensions—11.375" high; 16.250" wide; 11.250" deep (exclude video input connector) • Weight—6.5 KG (14.3 lbs) net

Use Master Charge/Visa or send money order.

Micro Products Unlimited
P.O. Box 1525, Arlington, TX 76010
817/461-8043
Dealer inquiries welcome ✓176

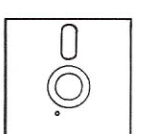
WE WILL NOT BE UNDERSOLD!!!

Find the best price you can in this magazine on a box of 10, 5¼" Verbatim diskettes and

Subtract 50¢

(Low discount price—$26.50)

THAT'S OUR PRICE



Alpha Byte Storage ✓134

4636 Park Granada 159
Calabasas California
91302 (213) 992-1970

**Offer good while supply lasts. Price includes shipping in the U.S. except for Alaska and Hawaii.*

'68' MICRO JOURNAL™

★ The only ALL **6800** Computer Magazine.

Foreign Orders—Add:
Air Mail $29.00/Year Surface $9.50/Year

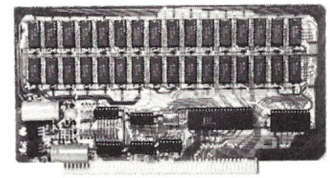
1-Year $14.50 2 Years $26.00
3 Years $36.50

OK, PLEASE ENTER MY SUBSCRIPTION

Bill my: M/C ☐ — VISA ☐
Card #__________
Expiration Date__________
For ☐ 1-Yr. ☐ 2 Yrs. ☐ 3 Yrs.
Enclosed: $__________
Name__________
Street__________
City__________
State__________ Zip______

'68' MICRO JOURNAL™
3018 Hamill Road
HIXSON, TN 37343 ✓132

The days of complicated, unreliable, dynamic RAM are gone:



INTRODUCING

JAWS

the ultrabyte memory board

$199.95 (*complete kit with 16K memory*)

Netronics consistently offers innovative products at unbeatable prices. And here we go again — with JAWS, the ultrabyte 64K S100 memory board.

ONE CHIP DOES IT ALL

JAWS solves the problems of dynamic RAM with a state-of-the-art chip from Intel that does it *all*. Intel's single chip 64K dynamic RAM controller eliminates high-current logic parts . . . delay lines . . . massive heat sinks . . . unreliable trick circuits.

REMARKABLE FEATURES OF JAWS

Look what JAWS offers you: Hidden refresh . . . fast performance . . . low power consumption . . . latched data outputs . . . 200 NS 4116 RAMs . . . on-board crystal . . . 8K bank selectable . . . fully socketed . . . solder mask on both sides of board . . . designed for 8080, 8085, and Z80 bus signals . . . works in Explorer, Sol, Horizon, as well as all other well-designed S100 computers.

GIVE YOUR COMPUTER A BIG BYTE OF MEMORY POWER WITH JAWS — SAVE UP TO $90 ON INTRODUCTORY LIMITED-OFFER SPECIAL PRICES!

UNDECIDED? *TRY A WIRED 16K JAWS IN YOUR COMPUTER ON OUR 10-DAY MONEY-BACK OFFER (SPECIFY YOUR COMPUTER).*

CONTINENTAL U.S.A. CREDIT CARD BUYERS OUTSIDE CONNECTICUT CALL
CALL TOLL FREE 800-243-7428
From Connecticut Or For Assistance. (203) 354-9375

NETRONICS RESEARCH & DEVELOPMENT LTD.
333 Litchfield Road, New Milford, CT 06776

Please send the items checked below: Dept. K-5

- ☐ **JAWS 16K RAM kit**, No. 6416, $199.95.*
- ☐ **JAWS 16K RAM fully assembled, tested, burned in**, No. 6416W, $229.95.*
- ☐ **JAWS 32K RAM kit**, No. 6432, (reg. price $329.95), **SPECIAL PRICE $299.95.***
- ☐ **JAWS 32K RAM fully assembled, tested, burned in**, No. 6432W, (reg. price $369.95), **SPECIAL PRICE $339.95.***
- ☐ **JAWS 48K RAM kit**, No. 6448, (reg. price $459.95), **SPECIAL PRICE $399.95.***
- ☐ **JAWS 48K fully assembled, tested, burned in**, No. 6448W, (reg. price $509.95), **SPECIAL PRICE $449.95.***
- ☐ **JAWS 64K RAM kit**, No. 6464, (reg. price $589.95), **SPECIAL PRICE $499.95.***
- ☐ **JAWS 64K RAM fully assembled, tested, burned in**, No. 6464W, (reg. price $649.95), **SPECIAL PRICE $559.95.***
- ☐ **Expansion kit, JAWS 16K RAM module**, to expand any of the above in 16K blocks up to 64K, No. 16EXP, $129.95.*

*All prices plus $2 postage and handling. Connecticut residents add sales tax.

Total enclosed: $__________
☐ Personal Check ☐ Money order or Cashiers Check
☐ VISA ☐ MASTER CHARGE (Bank No. ____)
Acct. No. __________ Exp. Date ______
Signature__________
Print Name__________
Address__________
City__________
State__________ Zip______
☐ Send me more information

```
0900  7E E0 CC 7E E0 CA 7E E0 C8 7E E0 7E 7E E1 AC 7E
0910  E1 D1 A0 42 00 00 00 00 0A 7E 00 08 00 0A 00 00
0920  09 00 30 30 30 39 04 30 38 20 20 20 20 20 20 20
0930  20 49 4E 58 20 20 20 20 20 20 20 20 20 20 20 04
0940  00 00 00 00 00 00 00 00 00 00 8E A0 42 BF 09 12
0950  CE 0A E0 FF A0 14 BE 09 12 7F 09 21 CE 0B 8A BD
0960  09 09 BD 09 0C 16 BD 09 00 C1 44 27 18 C1 47 27
0970  7B C1 54 27 27 C1 52 27 06 C1 4D 27 05 20 D7 7E
0980  0A 11 7E 0A 17 BD 09 CE BD 0B A3 CE 09 22 BD 09
0990  09 CE 0B 9C BD 09 09 BD 0A 97 20 EC BD 09 CE 30
09A0  B6 09 17 A7 05 B6 09 18 A7 06 BD 0B A3 CE 09 22
09B0  86 04 B7 09 26 BD 09 09 CE 09 2F BD 09 09 BD 0A
09C0  BE CE 0B 9C BD 09 09 BD 0A 97 BD 0B 11 3B BD 0A
09D0  65 FF 09 17 BD 09 0C 81 0D 27 0A BD 0A 59 FF 09
09E0  1F 7C 09 21 39 CE 0B 9C BD 09 09 39 BD 0A 65 FF
09F0  09 15 30 B6 09 15 A7 05 B6 09 16 A7 06 7F 09 1E
0A00  BD 09 0C 81 0D 27 09 BD 0A 59 FF 09 17 BD 0B 11
0A10  3B 30 FF 09 15 20 03 BD 0A 65 CE 0B 9C BD 09 09
0A20  CE 09 15 BD 09 06 FE 09 15 BD 09 03 FF 09 15 BD
0A30  09 0C 81 20 27 E4 81 08 26 0A FE 09 15 09 09 FF
0A40  09 15 20 D6 BD 0A 81 BD 0A 75 09 A7 00 A1 00 27
0A50  C9 86 3F BD 09 0F 7E 09 56 8D 0A CE 0B 9C BD 09
0A60  09 FE 09 15 39 8D 0C B7 09 15 8D 07 B7 09 16 FE
0A70  09 15 39 8D 09 48 48 48 48 16 8D 02 1B 39 BD 09
0A80  0C 80 30 2B 0F 81 09 2F 0A 81 11 2B 07 81 16 2E
0A90  03 80 07 39 7E 09 56 7D 09 21 27 0D FE 09 17 BC
0AA0  09 1F 26 0A 7F 09 21 20 05 BD 09 0C 20 08 B6 80
0AB0  09 2A 0A B6 80 08 81 1B 26 03 7E 09 56 39 FE 09
0AC0  12 08 BD 09 03 BD 09 03 BD 09 03 BD 09 06 08 FF
0AD0  09 15 CE 09 15 BD 09 06 FE 09 15 08 BD 09 06 39
0AE0  BF 09 12 30 6D 06 26 02 6A 05 6A 06 8D 09 30 EE
0AF0  05 FF 09 17 7E 09 AA B6 09 1E 84 3C 81 08 26 08
0B00  FE 09 19 B6 09 14 A7 00 FE 09 17 B6 09 1B A7 00
0B10  39 B6 09 1E 84 1C 26 03 8D 32 39 81 04 26 08 FE
0B20  09 19 FF 09 17 20 F1 81 08 26 04 8D 53 20 E9 81
0B30  0C 26 11 4F F6 09 19 30 EB 06 A9 05 B7 09 17 F7
0B40  09 18 20 D4 30 EE 09 FF 09 17 20 CC FE 09 17 A6
0B50  00 B7 09 1B 86 3F A7 00 A1 00 27 23 CE 09 17 BD
0B60  09 06 CE 0B 92 BD 09 09 BD 0A 97 B6 09 1E 85 20
0B70  27 05 FE 09 1C 20 03 30 EE 0B FF 09 17 20 CD 39
0B80  FE 09 19 A6 00 B7 09 14 20 CA 0D 0A 00 00 00 00
0B90  3A 04 20 2A 2A 2A 20 20 20 52 4F 4D 0D 0A 00 00
0BA0  00 00 04 CE 09 22 C6 1D 86 20 A7 00 08 5A 26 FA
0BB0  86 04 A7 00 FE 09 17 A6 00 B7 09 1B 48 CE 0D F4
0BC0  FF 09 1C 24 03 7C 09 1C F6 09 1D 1B B7 09 1D 24
0BD0  03 7C 09 1C CE 09 22 B6 09 17 BD 0C FF B6 09 18
0BE0  BD 0C FF FE 09 1C A6 00 B7 09 1C E6 01 F7 09 1E
0BF0  C4 03 FE 09 17 A6 01 B7 09 19 A6 02 B7 09 1A 37
0C00  CE 09 27 B6 09 1B BD 0C FF 08 5A 27 0F B6 09 19
0C10  BD 0C FF 5A 27 06 B6 09 1A BD 0C FF 33 FE 09 17
0C20  08 5A 26 FC FF 09 17 B6 09 1C 16 CE 0D 19 FF 09
0C30  1C 48 24 04 7C 09 1C 0C 1B 24 03 7C 09 1C 0C BB
0C40  09 1D 24 03 7C 09 1C B7 09 1D FE 09 1C A6 00 B7
0C50  09 31 A6 01 B7 09 32 A6 02 B7 09 33 C6 20 F7 09
0C60  34 B6 09 1E 85 C0 27 08 2A 04 C6 41 20 02 C6 42
0C70  F7 09 35 CE 09 37 B6 09 31 81 2A 26 08 B6 09 1B
0C80  BD 0C FF 20 40 B6 09 1E 85 02 27 39 B6 09 1B 81
0C90  8D 27 12 84 F0 B7 09 1B 81 80 27 04 81 C0 26 05
0CA0  86 23 A7 00 08 B6 09 1B 81 8D 27 38 84 F0 81 20
0CB0  27 32 B6 09 19 BD 0C FF B6 09 1E 85 01 27 06 B6
0CC0  09 1A BD 0C FF B6 09 1B 81 60 27 08 81 A0 27 04
0CD0  81 E0 26 09 86 2C A7 00 08 86 58 A7 00 FE 09 17
0CE0  FF 09 1C 39 4F F6 09 19 2A 01 4A 0C FB 09 18 F7
0CF0  09 1A B9 09 17 B7 09 19 8D 05 17 8D 02 20 DE 36
0D00  8D 06 32 08 8D 06 08 39 44 44 44 44 84 0F 8B 30
0D10  81 39 23 02 8B 07 A7 00 39 2A 2A 2A 4E 4F 50 54
0D20  41 50 54 50 41 49 4E 58 44 45 58 43 4C 56 53 45
0D30  56 43 4C 43 53 45 43 43 4C 49 53 45 49 53 42 41
0D40  43 42 41 54 41 42 54 42 41 44 41 41 41 42 41 42
0D50  52 41 42 48 49 42 4C 53 42 43 43 42 43 53 42 4E
0D60  45 42 45 51 42 56 43 42 56 53 42 50 4C 42 4D 49
0D70  42 47 45 42 4C 54 42 47 54 42 4C 45 54 53 58 49
0D80  4E 53 50 55 4C 44 45 53 54 58 53 50 53 48 52 54
0D90  53 52 54 49 57 41 49 53 57 49 4E 45 47 43 4F 4D
0DA0  4C 53 52 52 4F 52 41 53 52 41 53 4C 52 4F 4C 44
0DB0  45 43 49 4E 43 54 53 54 43 4C 52 4A 4D 50 53 55
0DC0  42 43 4D 50 53 42 43 41 4E 44 42 49 54 4C 44 41
0DD0  45 4F 52 41 44 43 4F 52 41 41 44 44 43 50 58 42
0DE0  53 52 4C 44 53 53 54 41 53 54 53 4A 53 52 4C 44
0DF0  58 53 54 58 00 01 01 01 00 01 00 01 00 01 00 01
0E00  02 01 03 01 04 01 05 01 06 01 07 01 08 01 09 01
0E10  0A 01 0B 01 0C 01 0D 01 0D 01 00 01 00 01 00 01
0E20  0E 01 0F 01 00 01 10 01 00 01 11 01 00 01 00 01
0E30  00 01 00 01 12 06 00 01 13 0A 14 0A 15 0A 16 0A
0E40  17 0A 18 0A 19 0A 1A 0A 1B 0A 1C 0A 1D 0A 1E 0A
0E50  1F 0A 20 0A 21 01 22 01 23 81 23 41 24 01 25 01
0E60  26 81 26 41 00 01 27 11 00 01 28 11 00 01 00 01
0E70  29 01 2A 01 2B 81 00 01 00 01 2C 81 2D 81 00 01
0E80  2E 81 2F 81 30 81 31 81 32 81 00 01 33 81 34 81
0E90  00 01 35 81 2B 41 00 01 00 01 2C 41 2D 41 00 01
0EA0  2E 41 2F 41 30 41 31 41 32 41 00 01 33 41 34 41
0EB0  00 01 35 41 2B 02 00 01 00 01 2C 02 2D 02 00 01
0EC0  2E 02 2F 02 30 02 31 02 32 02 00 01 33 02 34 02
0ED0  36 0E 35 02 2B 03 00 01 00 01 2C 03 2D 03 00 01
0EE0  2E 03 2F 03 30 03 31 03 32 03 00 01 33 03 34 03
0EF0  36 07 35 03 37 82 38 82 39 82 00 01 3A 82 3B 82
0F00  3C 82 00 01 3D 82 3E 82 3F 82 40 82 41 03 42 26
0F10  43 03 00 01 37 82 38 82 39 82 00 01 3A 82 3B 82
0F20  3C 82 44 82 3D 82 3E 82 3F 82 40 82 41 02 00 01
0F30  43 02 45 02 37 82 38 82 39 82 00 01 3A 82 3B 82
0F40  3C 82 44 82 3D 82 3E 82 3F 82 40 82 41 02 46 2E
0F50  43 02 45 02 37 83 38 83 39 83 00 01 3A 83 3B 83
0F60  3C 83 44 83 3D 83 3E 83 3F 83 40 83 41 03 46 27
0F70  43 03 45 03 37 42 38 42 39 42 00 01 3A 42 3B 42
0F80  3C 42 00 01 3D 42 3E 42 3F 42 40 42 00 01 00 01
0F90  47 03 00 01 37 42 38 42 39 42 00 01 3A 42 3B 42
0FA0  3C 42 44 42 3D 42 3E 42 3F 42 40 42 00 01 00 01
0FB0  47 02 48 02 37 42 38 42 39 42 00 01 3A 42 3B 42
0FC0  3C 42 44 42 3D 42 3E 42 3F 42 40 42 00 01 00 01
0FD0  47 02 48 02 37 43 38 43 39 43 00 01 3A 43 3B 43
0FE0  3C 43 44 43 3D 43 3E 43 3F 43 40 43 00 01 00 01
0FF0  47 03 48 03 00 00 00 00 00 00 00 00 00 00 00 00
>
```

Hex listing of Trace and Disassemble program.

monitor.

Program B shows the disassembly of Tiny BASIC starting at address 0225 hex and finishing at 0280 hex. All values are in hexadecimal. Branch operands are the actual branch address. Direct addressing instructions are shown with two digit operands. If a location does not contain a valid op code, the disassembler will assume it is data and output asterisks (***) for the mnemonic.

Memory Examine and Change Function

This function can be used by the operator for inputting a program or making changes to an existing program. This function can be performed by typing in the following response:

: M nnnn

The character M specifies a memory change function. The four hexadecimal digits following M specify the address to be examined or changed. Once the last digit is entered, the program will respond with the address and its contents:

: M 0103
0103 7E

The operator must now decide whether to change memory, space to the next location, back space to the previous location or return to the control monitor.

If the contents of memory are to be changed, just enter the new value. The program will automatically output the next address and its contents. If the contents of memory cannot be changed, the program will output a (?) and return to the control monitor.

If the operator wishes to space to the next location, he'll just depress the space bar. The program will output the next location and its contents. For back spacing to the previous location, just depress the back space key (08 hex). The program will output the previous location and its contents. The back space function is useful for back spacing when an incorrect value is entered.

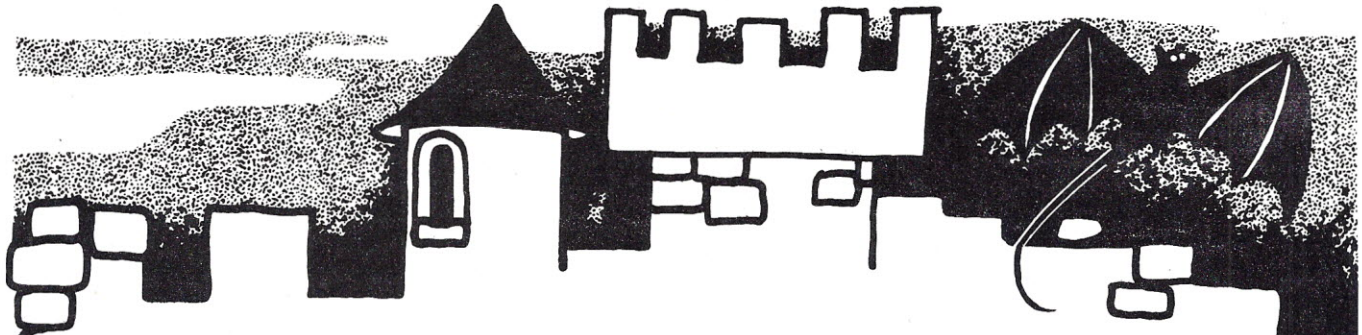
The memory change function can be terminated by depressing the Escape key or entering an invalid hex character (see Example 6).

Register Examine and Change Function

This function is used to ex-

```
: M 0103
0103   7E          (space)
0104   02          (back space)
0103   7E    BD
0104   02          (back space)
0103   BD    7E
0104   02          (space)
0105   25          (back space)
0104   02          (back space)
0103   7E          (escape)
```

Example 6.



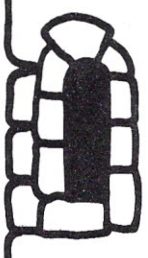
Did you read about the Dungeonmaster who became so enchanted playing a real life version of Dungeons and

Dragons that he disappeared for a month?

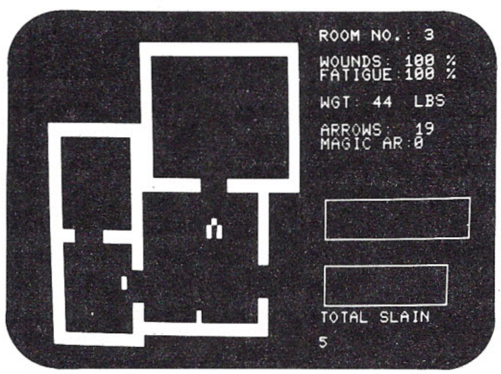
You'll be able to hold on to reality just a little better when you play the Dunjonquest™ computer version, the greatest of all the role-playing fantasies.

But don't bet on it.

Sit at your computer. You're the hero. Enter the Dunjonquest "Temple of Apshai" and into the greatest fantasy adventure you've ever experienced. The Temple has over 200 rooms and catacombs in which lurk more than 30 kinds of monsters and beasts ready to do you in—in real time—before you can reach any of the 70 or so treasures waiting for the hero. You may spend days, weeks, months... the rest of your life... striking at the forces of evil, or running from them, or calling on powers you can never completely understand. Always, always demonstrating in varying degrees your strength, constitution, dexterity, intelligence, intuition, the force of your ego.



Unlike chess or bridge or monopoly, this role-playing game—like other good role-playing games—is an **experience** rather than a game: it is not played so much as it's lived or experienced. Your alter ego goes forth into the world of demons and darkness, dragons and dwarves. Your character will do whatever you want him (or her or it) to do.



Actual photo of screen during a Dunjonquest game. In Room 3 in the Temple of Apshai, our hero observes two treasures unattended by dragons, monsters or demons... for the moment. He is completely free of wounds; he is not at all fatigued. He carries 44 pounds of armor and 19 arrows in his quiver. He has already slain five demons. Will he capture the treasures before moving on... or before the forces of darkness intercept him?

"The Temple..." comes complete with a superbly illustrated 56-page rule book and cassette program, designed to operate with the Level II 16K TRS 80, the PET 32K or the Apple II 48K (Applesoft) computer. Only $24.95 complete, **including** shipping and handling on orders placed within the next 30 days. (Apple or TRS 80 disk available for $29.95).

Dunjonquest's "The Temple of Apshai" is **guaranteed** to be the best version of Dungeons and Dragons/Dragons and Dungeons. It's a product of the two guys who **are** Automated Simulations: Jim Connelley and Jon Freeman. Jim is a Dungeon Master, running continuous D & D campaigns. He's been a data processing professional with Westinghouse, GTE Sylvania, Logisticon... an expert in computer-based math-modeling and in simulation of complex phenomena. Jon is a game player, designer and author. He's a frequent contributor to *Games* magazine; his book, "The Playboy Winner's Guide to Board Games" is a paperback best-seller.

As we said, **guaranteed**: Guaranteed to be the best version; guaranteed that you'll be happy with it. Order now, use it for two weeks. If you don't enjoy completely this fantasy adventure experience that goes beyond all others, send it back to us. We'll refund your money in full; no questions asked.

Master Charge or Visa card holders: charge "The Temple of Apshai" to your credit card. Just call our toll free number: (800) 824-7888, operator 861 (In California, call operator 861 (800) 852-7777. In Hawaii and Alaska, operator 861 (800) 824-7919) and you can begin enjoying your Dunjonquest game in days. Or send your check for $24.95* (or $29.95)* to

✓282

Automated Simulations

Dept. K-A
P.O. Box 4232
Mountain View,
CA 94040

*California residents, please add 6.5% tax.

```
MPU Register   : R
cc             A077 C1        (space)
B              A078 19   FE
A              A079 0D   A0
XH             A07A 22        (space)
XL             A07B 42        (space)
PCH            A07C 01        (space)
PCL            A07D 03        (space)
RTNH           A07E 02        (space)
RTNL           A07F 2A        (space)
               A080 FF        (escape)
               : T 0103
               0103   JMP   0225   C1   FE   A0   2242   A07D   022A
```

Example 7.

amine and change the contents of the MPU registers prior to executing the trace or Go to User's Program function. The trace and Go to User's Program functions use the return from interrupt (RTI) instruction to return to the user's program. The RTI instruction updates all the MPU registers with the values stored away in the stack.

The register examine and change function is initiated by entering the character R after the colon. The location of the first MPU register and its contents will be printed. The examining and changing of the data is done in the same manner as the M function (see Example 7).

Basic Memory Map

The 6800 Trace and Disassemble program resides in less than 2K of memory. The hex listing accompanies the article. The program uses some of the MIKBUG I/O routines. Table 1 lists I/O routines used by the program.

There are some parameters that may have to be changed depending on your particular machine. The stack pointer, for example, is initially loaded to $A042. If this value is changed, it should be set to at least ten locations down from the top of the stack.

The software interrupt vector is normally stored at location $FFFA. In my home-brew system the software interrupt vector points to a ROM subroutine that uses location $A014 as a programmable software interrupt vector. The Trace and Disassemble program initializes location $A014 to the return address of the trace function. This address ($A014) in the program will have to be changed to $FFFA (if programmable) or to whatever the programmable location is in your particular machine.

The Back Space and Escape Codes can be modified. They are presently set to 08 hex and 1B hex, respectively.

Break Test Routine

The break test is used by the program during a trace or disassemble program function. After each line of output the program jumps to the break test routine. The break test checks for a key being depressed. If one is not, the program returns normally. If a key is depressed, the character is input and tested for the Escape Code. If the character is not the Escape Code, the program exits from the routine normally. If the character is the Escape Code, the program returns to the control monitor.

Any changes to the break test must be made within the first three instructions. The remaining four are used by other routines within the program. There are some spare locations at the end of the program starting at $0FF4 for modifications to the break test (see Example 8).

Summary

The 6800 Trace and Disassemble program is an effective debugging tool. It requires no hardware changes, as long as your system has a programmable SWI vector. I've used it many times and so have other 6800 users. It allows you to trace your program instruction by instruction. You can make changes to your program, disassemble your patches and then trace them. You can make a listing of your program and even the trace of your program.

If you would like to get a copy of the listing of the program for relocation purposes or whatever, just send $5 with your name and address to:

Richard Carickhoff
812 Pulaski Drive
Lansdale PA 19446

If you have any problems with the program just send a self-addressed, stamped envelope to me and I'll try to answer any questions that you may have. ■

	JMP $E1AC	OUTPUT 2 HEX CHARS AND SPACE
BASIC MEMORY MAP		
0900-0911	I/O ROUTINES	
0912-0949	TEMPORARY STORAGE	
094A	START OF PROGRAM	
094A-0D18	EXECUTABLE PROGRAM	
0B8A-0BA2	PROMPT, INVALID CODE AND CRLF MESSAGES	
0D19-0FF3	MNEMONIC AND CODE TABLES	
MIKBUG I/O ROUTINES		
0900	JMP $E0CC	OUTPUT SPACE
0903	JMP $E0CA	OUTPUT 2 HEX CHARS AND SPACE
0906	JMP $E0C8	OUTPUT 4 HEX CHARS AND SPACE
0909	JMP $E07E	OUTPUT MESSAGE
090C	JMP $E1AC	INPUT A CHAR
090F	JMP $E1D1	OUTPUT A CHAR
PARAMETERS		
094B-094C	$A042	MIDDLE OF STACK
0954-0955	$A014	SWI VECTOR (NORMALLY $FFFA)
0A37	$08	BACKSPACE CODE
0AB7	$1B	ESCAPE CODE

Table 1. Memory map of I/O routines and parameters.

T nnnn (CR)	Trace instruction at location nnnn.
T nnnn, mmmm	Trace program starting at location nnnn with breakpoint address set at mmmm.
G nnnn (CR)	Go to user's program starting at location nnnn.
G nnnn, mmmm	Go to user's program starting at location nnnn with breakpoint address set at mmmm.
D nnnn (CR)	Disassemble instruction at location nnnn.
D nnnn, mmmm	Disassemble instruction at location nnnn and ending at location mmmm.
M nnnn	Examine memory location nnnn.
R	Examine MPU registers starting with condition code.
(ESC)	Escape from present function and return to control monitor.

Table 2. Summary of control functions.

```
0AAE   BREAK   LDAA   $8009     PIA STATUS-KEY DEPRESSED?
0AB1           BPL    EXIT      NO
0AB3           LDAA   $8008     YES, INPUT CHAR
0AB6   CHECK   CMPA   #1B       ESCAPE CODE?
0AB8           BNE    EXIT      NO
0ABA           JMP    CONTROL   YES, RETURN TO CONTROL MONITOR
0ABD   EXIT    RTS              RETURN NORMAL
```

Example 8.

Programming Optimization Techniques

If you want to optimize a program, use the best approach—or don't bother.

W. A. Harrison
Computer Science Dept.
University of Missouri
Rolla MO 65401

The most important step in coding an efficient program is to make sure that the procedure used is the best approach to the problem. Simply coding the best method, without any attempt to optimize the program, will generally result in a more efficient program than trying to optimize a second- or third-best approach.

Optimization of a computer program is usually done to accomplish one of two results: increased speed or decreased storage. In some cases, an optimizing technique can take on both aspects, and yet, others require a trade-off to be made in terms of speed and memory. Our discussion in this article shall be limited to those tech niques that increase the execution speed of a program.

Before we decide to attempt to optimize a program, we must first decide if our program is worth the effort required. Though you should always strive to code the most efficient program possible, some programs are not worth the effort it requires. An example of this is a short program that will only be used once or twice. On the other hand, a large program that involves a great deal of number-crunching, and is utilized quite often, may well be worth the effort required to optimize the code.

One of the most obvious methods of optimizing a computer program is by minimizing the number of operations the machine must perform. Yet this is probably the single largest reason for program inefficiency. This is doubly important when an interpreter, such as BASIC, is used, which translates and then executes each source language statement as it is encountered, as opposed to a compiler (e.g., FORTRAN), which translates the entire source program first, and then executes the resulting machine code.

One of the largest areas of inefficiency in most programs is the placement of calculations within loops whose outcome is not affected by the operations taking place within the loop. In the program segment in Example 1, the expression C + R is executed 1000 times. The result, however, never varies. The program segment could be coded much more efficiently by placing the C + R calculation outside the loop and only calculating it once. This will result in the elimination of 999 unneeded evaluations of the expression C + R.

Likewise, any redundant expressions should be computed once, and the result stored in a temporary storage location to allow the machine to avoid repeating the calculation. In Example 2, the expression C + R is calculated several times. This segment could be more efficiently coded by setting the variable T equal to the results of C + R and using this value in the computations to avoid computing the expression over and over again.

Often, an expression whose value never changes from run to run will appear in a program. In Example 3, the value of X is always 3. Rather than requiring the machine to calculate this value every time the program is

inefficient
```
00100  FOR K=1 TO 1000
00110        B=C+R
00120        Q1=Q(K)B
00100  NEXT K
```
efficient
```
00100  B=C+R
00110  FOR K=1 to 1000
00120        Q1=Q(K)/B
00130  NEXT K
```
Example 1.

inefficient
```
00100  R=(C+R)/A
00110  S=(C+R)*A
00120  U=C+R-((C+R)*A)
```
efficient
```
00100  T=C+R
00110  R=T/A
00120  S=T*A
00130  U=T-(T*A)
```
Example 2.

inefficient
```
00100  X=LGT(1000)
```
efficient
```
00100  X=3
```
Example 3.

inefficient
```
00100  Q=(-C*100)
```
efficient
```
00100  Q=(-100*C)
```
Example 4.

```
      inefficient                        efficient
00100  FOR C=1 TO 100             00100  FOR C=2 TO 200 STEP 2
00110       A(C*2)=K              00110       A(C)=K
00120  NEXT C                     00120  NEXT C
```

Example 5.

```
   inefficient
00100  FOR I=1 TO 100                I is set to 1 1 time
00110       FOR J=1 TO 50            J is set to 1 100 times
00120            FOR K=1 TO 10       K is set to 1 5000 times
                                   result: 5101 initializations

   efficient
00100  FOR K=1 TO 10                 K is set to 1 1 time
00110       FOR J=1 TO 50            J is set to 1 10 times
00120            FOR I=1 TO 100      I is set to 1 500 times
                                   result: 511 initializations
```

Example 6.

```
        inefficient                        efficient
00100  IF SQR(K) = N THEN 1000     00100 IF K = N*N THEN 1000
```

Example 7.

```
   inefficient          more efficient           most efficient
00100 M=C(K)+8       00100 M=C(3)+8           00100 T=C(3)
00110 N=C(K)*8             •                  00110 M=T+8
      •                    •                        •
      •                    •                        •
      •                    •                        •
00150 Z=C(K)/2       00150 Z=C(3)/2           00150 Z=T/2
```

Example 8.

```
   inefficient                              efficient
00100 FOR K=1 TO N                       00100 A(N+1)=V
00110      IF A(K)=V THEN 150            00110 K=0
00120 NEXT K                             00120 K=K+1
00130 PRINT "VALUE NOT FOUND"            00130 IF A(K)=V THEN 150
00140 STOP                               00140 GOTO 120
                                         00150 IF K<=N THEN 180
                                         00160 PRINT "VALUE NOT FOUND"
                                         00170 STOP
```

Example 9.

executed, X can be set to 3, eliminating the computation.

A less obvious technique, yet one that can result in a small savings in time, is the proper positioning of a unary operator, such as the minus sign, preceding a variable. In Example 4, the operator "-" acting on the variable C requires the equivalent of C*(−1). However, this segment could be programmed in a more efficient way that would eliminate the unary operation of (−1)* (the value −100 is a constant and does not require a unary operation to make it negative).

Efficiency can also be obtained by making full use of the FOR/NEXT increment parameters as shown in Example 5. This will eliminate the computation of C*2 at every iteration of the loop.

Often, small things can be overlooked by the programmer, yet they can make a significant savings when applied enough times. For example, every time a FOR/NEXT statement is executed, the loop counter must be initialized to the beginning value (in Example 6, each counter must be initialized to 1 at the beginning of the loop). By careful arrangement of the nested loops, the number of initializations may be reduced.

As we stated before, execution time is reduced when we can reduce the amount of work the machine must do. One very time-consuming operation involves locating and retrieving a system library subroutine. In Example 7, the result is the same if a number is multiplied by itself (squared), or if the number it is being compared to has its square root computed. However, it takes less time to multiply two numbers together than to compute a square root.

In our quest for efficiency, we must also take into consideration how long it takes the machine to locate and retrieve the contents of a storage location referenced by a variable. In the case of a scalar (non-subscripted) variable, the system must consult a table, from which the address of that location is found. On the other hand, when a subscripted variable is referenced, the system must find the starting point of the array or vector, and then using the subscript, calculate the address of the storage location in question.

Because of this, when a subscripted variable is referenced several times, efficiency may be gained by setting a scalar variable equal to the subscripted variable and using the scalar variable for access. If this cannot be done, access time can still be increased by using a constant as a subscript rather than a variable (i.e., V(7) instead of V(K)). This is because when a variable is used as a subscript, the contents of the location referenced by the variable must be computed. In effect, two storage locations must be located and retrieved. See Example 8.

Probably the most time-consuming operation carried out by a computer is the decision operation. Because of this, the use of IF statements should be minimized whenever possible. Through commonsense programming, you can often eliminate, if not the use of, at least the execution of, a number of IF statements (e.g., positioning the most likely test first in order to branch around a number of other IF statements when testing a value for a number of conditions).

The programmer must also keep in mind that the IF statement is not the only operation that requires a decision-making operation. There are several operations that require implicit condition testing. An example is the FOR/NEXT statement. The counter is checked every time a loop is executed to determine if the iteration should continue, or if the loop should be terminated. Because of this, the FOR/NEXT statement is not always the most desirable method of performing a loop.

In Example 9, up to 2N decisions must be made to determine if the value being sought, V, is in the array A. A more efficient method would be to eliminate the FOR/NEXT operation as shown in the example. This would require only up to N+2 decision operations.

Of course, the amount of time saved by utilizing the preceding techniques may be slight, but if they are used consistently, especially in those huge number-crunchers referred to earlier, the savings in computer time can be considerable. ■

Bibliography

Harry Katzan, Jr., *Computer Systems Organization and Programming*, Science Research Associates, Inc., Chicago IL, 1976.

Robert E. Lynch and John R. Rice, *Computers: Their Impact and Use*, Holt, Rinehart and Winston, New York, 1977.

Daniel D. McCracken, *A Guide to Fortran IV Programming*, John Wiley & Sons, Inc., 1972.

Robert and Nancy Stern, *BASIC Supplement to Accompany Principles of Data Processing*, John Wiley & Sons, Inc., 1974.

THE ORIGINAL MAGAZINE FOR OWNERS OF THE TRS-80™* MICROCOMPUTER

SOFTWARE FOR TRS-80™ OWNERS

H & E COMPUTRONICS INC.

MONTHLY NEWSMAGAZINE FOR TRS-80™ OWNERS

MONTHLY NEWSMAGAZINE
Practical Support For Model I & II

- PRACTICAL APPLICATIONS
- BUSINESS
- GAMBLING • GAMES
- EDUCATION
- PERSONAL FINANCE
- BEGINNER'S CORNER
- NEW PRODUCTS
- SOFTWARE EXCHANGE
- MARKET PLACE
- QUESTIONS AND ANSWERS
- PROGRAM PRINTOUTS

. . . . AND MORE

PROGRAMS AND ARTICLES PUBLISHED IN OUR FIRST 12 ISSUES INCLUDE THE FOLLOWING:

- A COMPLETE INCOME TAX PROGRAM (LONG AND SHORT FORM)
- INVENTORY CONTROL
- STOCK MARKET ANALYSIS
- WORD PROCESSING PROGRAM (FOR DISK OR CASSETTE)
- LOWER CASE MODIFICATION FOR YOUR VIDEO MONITOR OR PRINTER
- PAYROLL (FEDERAL TAX WITHHOLDING PROGRAM)
- EXTEND 16-DIGIT ACCURACY TO TRS-80™ FUNCTIONS (SUCH AS SQUARE ROOTS AND TRIGONOMETRIC FUNCTIONS)
- NEW DISK DRIVES FOR YOUR TRS-80™
- PRINTER OPTIONS AVAILABLE FOR YOUR TRS-80™
- A HORSE SELECTION SYSTEM***ARITHMETIC TEACHER
- COMPLETE MAILING LIST PROGRAMS (BOTH FOR DISK OR CASSETTE SEQUENTIAL AND RANDOM ACCESS)
- RANDOM SAMPLING***BAR GRAPH
- CHECKBOOK MAINTENANCE PROGRAM
- LEVEL II UPDATES***LEVEL II INDEX
- CREDIT CARD INFORMATION STORAGE FILE
- BEGINNER'S GUIDE TO MACHINE LANGUAGE AND ASSEMBLY LANGUAGE
- LINE RENUMBERING
- AND CASSETTE TIPS, PROGRAM HINTS, LATEST PRODUCTS COMING SOON (GENERAL LEDGER, ACCOUNTS PAYABLE AND RECEIVABLE, FORTRAN 80, FINANCIAL APPLICATIONS PACKAGE, PROGRAMS FOR HOMEOWNERS, MERGE TWO PROGRAMS, STATISTICAL AND MATHEMATICAL PROGRAMS (BOTH ELEMENTARY AND ADVANCED) . . . AND

FREE

WORD PROCESSING PROGRAM
(Cassette or Disk)

For writing letters, text, mailing lists, etc., with each new subscriptions or renewal.

LEVEL II RAM TEST
(Cassette or Disk)

Checks random access memory to ensure that all memory locations are working properly.

DATA MANAGEMENT SYSTEM
(Cassette or Disk)

Complete file management for your TRS-80™

CLEANUP
(Cassette or Disk)

Fast action Maze Game.

FREE

* TRS-80™ IS A TRADEMARK OF TANDY CORP.

SEND FOR OUR NEW 48 PAGE SOFTWARE CATALOG (INCLUDING LISTINGS OF HUNDREDS OF TRS-80™ PROGRAMS AVAILABLE ON CASSETTE AND DISKETTE). $2.00 OR **FREE** WITH EACH SUBSCRIPTIONS OR SAMPLE ISSUE.

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™ ✓6

Box 149 **New City, New York 10956**

master charge

24 HOUR ORDER LINE
(914) 425-1535

VISA

ONE YEAR SUBSCRIPTION $24 . . . .

TWO YEAR SUBSCRIPTION $48 . . . .

SAMPLE OF LATEST ISSUE $ 4 . . . .

START MY SUBSCRIPTION WITH ISSUE . . . .

(#1 - July 1978 • #7 - January 1979 • #12 - June 1979 • #18 - January 1980)

NEW SUBSCRIPTION . . . . RENEWAL . . . .

CREDIT CARD NUMBER ____________________ EXP. DATE__________

SIGNATURE____________________

NAME ____________________

ADDRESS ____________________ STATE__________ ZIP__________

***** ADD $6/YEAR (CANADA, MEXICO) - ADD $12/YEAR AIR MAIL - OUTSIDE OF U.S.A., CANADA & MEXICO *****

WHAT DO THEY SAY?

What do the reviewers say about SWORDQUEST and DEATH PLANET?

SWORDQUEST for the PET

"For those who want the very best."
– Recreational Computing

"The only fantasy simulation I have seen with graphics and animation."
– Len Lindsay, Pet Gazette

"Challenging and Exciting"
– Kilobaud Microcomputing

"The SWORDQUEST fever is once again at its peak."
– R. Zamora, Editor, Recreational Computing

You seek the jewels and holy sword in the tomb of a long-dead wizard.
SWORDQUEST cassette and instruction booklet $12.95

ESCAPE from the DEATH PLANET (with SOUND) for the PET

"Exciting Game"
– Popular Mechanics

"Fantasy Games Software has done it again . . . an exciting action game with good graphics, sound effects and real time programming. Graphics include animated laser fire and explosions."
– Kilobaud Microcomputing

Killer Robots and Imperial Stormtroopers search for you. Laser pistol combat inside a giant space station.
DEATH PLANET cassette and instruction booklet $12.95

Programs will run on any 8K or larger PET. Each cassette has both OLD ROM and NEW ROM versions.

Fantasy Games Software
P.O. Box 1683
Madison, Wisconsin 53701 ✓271

Apple II Software (TAKE 15% OFF LIST PRICE)

Master Catalog (Programma)—File Manager for all your disks. Catalogs all your disks on one, easy access file. A great filing system for all your disk based software $24.95

Tranquility Base (Stoneware)—Super fast Hi-Res arcade type lunar lander. Miss your landing and watch the LEM explode into pieces and fly out into space. (Requires disk, written in machine code) $24.95

FS1 Flight Simulator—3-D animation and cockpit view with full flight instrumentation. Learn to fly a realistic plane simulator and engage in aerial dogfights! (Sublogic, on cassette in assembly language) $24.95

Micro League Baseball (Stoneware)—Just like the fancy arcade baseball only better. In full color, this unbelievably unique program actually literally talks back to you. Hours and hours of fun $24.95

Apple Invaders (Programma)—The arcade invader game that has made millions of fans. Shoot down the advancing lines of invaders. Fast execution. Hi-Res graphics. (In machine code) Specify cassette ($15.95) or disk ($19.95)

MIGHTY BYTE COMPUTER INC.
P.O. Box 213
HO-HO-KUS, NEW JERSEY 07423
(201) 445-8256 ✓205

We carry a full line of Programma & Muse Software.
Visa & Mastercard.
Add $1 shipping for each 2 programs.
Atari Systems and Mattel Intellivision available for immediate delivery.

TRS-80-LEVEL II and Disk Programs

MULTIPLE REGRESSION 2.0—A disk based package of chained programs that permits model estimation using thousands of observations, user specified transformations (write them in BASIC during execution), X-Y plots, formatted for screen or printer + all features of Multiple Regression 1.0 .... $45.00

LEVEL II 16K PROGRAMS

Multiple Regression 1.0 .... $29.95
Linear Programming $29.95
0-1 Programming $29.95
Transportation Algorithm $29.95
Heuristic Line Balancing $29.95
Stat. Pack—medium, mode, mean (avg., harmonic, geometric), variance, histograms, Tests (T,X^2,F,) one variable regression, one and two-way ANOVA .. $9.95
Differential equations—6 methods $29.95
Queuing Statistics $14.95

LOWERCASE MOD—Includes excellent documentation + all parts (nothing else to buy), compatible with Electric Pencil $14.95

Available in Disk add $5
S.C. residents add 4% sales tax
Overseas orders add $5 for shipping

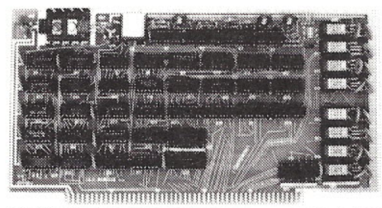
✓137

Quant Systems
p.o. box 628
charleston sc
29402

COLOR GRAPHICS FOR THE OHIO SCIENTIFIC CHALLENGER IP OR SUPERBOARD II

Details on modifications that will enable you to operate in several color graphic modes (8 colors plus black). Easy installation. Includes colored-video game for demonstration and checkout that will run on systems with 4K RAM. Low-cost hardware not included. Send $8.65 to

T. J. Hoffman ✓273
873 Dorset Drive
Knoxville, TN 37919



¿COMPUPRISM? COLOR

GRAPHICS FOR THE S 100 BUS. 16K OF ON BOARD MEMORY CAN BE USED AS RAM. 2 OR 4 MHz OPERATION. HIGH RESOLUTION (144 H. BY 192 V. PIXELS) WITH 16 COLORS AT THE SAME TIME. NO ADDRESS JUMPS, MAKES PROGRAMMING EASY. SOCKETS FOR ALL I.C.'S. KIT $240 A AND T $280

Bare board with documentation **$40.**

ALL ORDERS SHIPPED COD WITHIN 72 HOURS
4 MHz MOD FOR S.D. SYSTEMS EXPANDORAM $10
16 CHANNEL A-D, 8 CHANNEL D-A FOR S-100 BUS, BARE BOARD WITH DOCUMENTATION $25

J.E.S. GRAPHICS
P.O. BOX 2752
TULSA, OK. 74101 ✓180
(918) 742-7104

Available Direct From

DI$COUNT COMPUTER PRODUCT$

Master Charge & Visa Accepted
(Please give card no. and expiration date)

APPLE	
Apple II 16K	$999.
Micro Music Board for Apple	168.
OHIO SCIENTIFIC	
Superboard II	259.
C1P 4K	329.
C1P 8K	359.
C4P 8K	598.
C4P MF	1498.
C8P	795.
C8P DF	2349.
C2 OEM	2399.
NOVATION INC.	
Cat Modem	179.

TEXAS INSTRUMENTS	
Model 810 Basic Printer	1649.
CENTRONICS	
730-1	799.
779-2 Tractor Feed Printer	1049.
Epson Tx-80 Tractor Printer	688.
COMPRINT COMPUTER PRINTER INTERNATIONAL	
Comprint 912 S	598.
HAZELTINE	
1410	798.
1420	898.
1500	998.

MAIL & PHONE ORDERS ONLY! ✓199
SHIPPING EXTRA
DELIVERY FROM STOCK TO 6 WEEKS
P.O. BOX 308
Thiells, N.Y. 10984 (914) 429-9631

TRS-80 SPEEDUP BOARD REVERSE VIDEO

SPEED MOD — You don't have to spend $3,500 on a TRS-80 Model II to get faster computing. Now you can speedup your Level II TRS-80 - Disk systems included! - by up to 100% (50% guaranteed) with our speedup board. The result is more animated graphics, shorter program run times, and generally far greater computing power for your dollar. Change between normal and faster operation by using a simple BASIC statement. The contents of memory are not affected by speed changes and a switch is not required (a switch may be installed if manual speed select is desired). Changes are provided for NEWDOS, DOS 2.2, and DOS 2.3 that allow disk systems to run reliably at both the normal and accelerated rate. Buy the most versatile, easiest to install, and most publicly recognized speed mod on the market today.

ASSEMBLED & TESTED $24.95

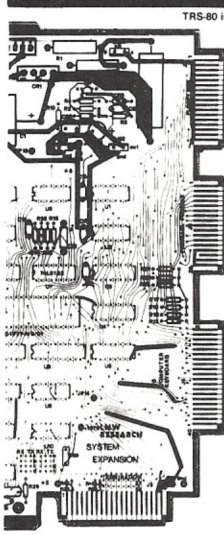
REVERSE VIDEO is finally here! If you're tired of going blurry-eyed looking at your video display, then you are ready for reverse video. It provides dark black characters and graphics on an all white screen for a much crisper and much easier to read presentation. Change between normal and reverse by simultaneously pressing a combination of three keys on the keyboard.

ASSEMBLED $14.95 ✓211

Add 5% for postage and handling
California residents add 6% sales tax

Bill Archbold Electronics
Dept 8 • P.O. Box 7123 • Sacramento, CA 95826
(916) 362-3627

TRS-80 is a registered trademark of TANDY CORP.



SYSTEM EXPANSION FOR THE TRS-80™

AT $69.95 [PC BOARD & USER MANUAL]

- SERIAL RS232C/20mA I/O
- FLOPPY CONTROLLER
- 32K BYTES MEMORY
- PARALLEL PRINTER PORT
- DUAL CASSETTE PORT
- REAL-TIME CLOCK
- SCREEN PRINTER BUS
- ONBOARD POWER SUPPLY
- SOFTWARE COMPATIBLE
- SOLDER MASK, SILK SCREEN

LNW RESEARCH ✓198

8 Hollowglen St. Irvine CA
714-552-8946 92714

—— TO ORDER ——
P.O. Box 16216 Irvine CA 92713
Add $3 for postage and handling.
CA residents add 6% sales tax
Master Charge & VISA orders now accepted

COMPUTER ACTION ✓183

Send for our
FREE PRICELIST
with very low prices
on disks, tapes, and
many other items for
COMPUTER HOBBYISTS
AND BUSINESS USERS

Send name & addr. to
Computer Action
P.O. Box 119 ✓183
Brooklyn, N.Y. 11236

Software for the AIM 65

Unravel some mysteries of the AIM 65's monitor subroutines.

John D. Williams
8832 Boehning Ct.
Indianapolis IN 46219

With the wide variety of single-board computers on the market today, you might think that the introduction of yet another is hardly needed. But the AIM 65 by Rockwell fills a gap left by other single-board computers. Its many features provide capabilities beyond those of the usual single-board systems, yet it costs little more.

The AIM 65 is a versatile microcomputer. It is a 6502-based system with ASCII keyboard, 20 column thermal printer and 20 character alphanumeric display. The basic unit includes an 8K monitor/miniassembler, two parallel I/O ports and 1K of 2114 RAM. An additional 3K RAM, assembler and 8K Microsoft BASIC are available as on-board options.

Though little has been written about this system, it should be given serious consideration by anyone wishing to purchase a single-board computer. The purpose of this article is to briefly discuss the hardware aspects of the AIM 65 and then unravel a few of the mysteries surrounding the use of the monitor subroutines in programming. I hope it will also provide some insights into the use of the two parallel ports for control or interface applications.

Introduction

The AIM 65 has two buses: one is an expansion bus for adding additional memory; the other is an application bus. Part of this bus consists of two 8-bit parallel I/O ports with four control lines. These ports are controlled by a 6522 VIA.

The VIA (versatile interface adapter) also has an 8-bit shift register to convert data between parallel and serial formats, interrupt logic that can be polled and two 16-bit timers that can produce a single pulse or a series of pulses. One of the timers will be used as the basis of the real-time clock in the program. The parallel ports will provide the interface to the outside world.

The program was written to read voltage from a Valhalla True RMS Wattmeter. It will record the changes in voltage and the time of those changes. But its real usefulness lies in demonstrating how you can use several unique features of the AIM in ways not mentioned in the *User's Manual*.

Specifically, the program uses the monitor's text editor to store the original prompt and subsequent statements used in the printout. It also uses both parallel I/O ports to specify what information is wanted and then to collect it. Finally, a real-time clock is used with one of the VIA's timers as its basis.

```
<G>/
THE VOLTAGE IS
      VRMS-60HZ.
THE TIME IS
THIS PROGRAM WILL
MONITOR LINE VOLTAGE
AND RECORD THE
CHANGES. IT WILL
ALSO RECORD THE TIME
OF THE CHANGES.
  TO INITIALIZE THE
PROGRAM, ENTER THE
TIME IN THIS FASHION
      HR:MIN:SEC.
ALLOW TWO DIGITS PER
UNIT. THEN HIT F2 TO
START THE PROGRAM AT
THE RIGHT TIME.
```

Listing 1. Program text.

Monitor Subroutines

Listing 1 shows the text that is used. The text editor is used to store this section of the program. This feature is more convenient than stuffing consecutive memory locations with the proper ASCII codes. Simply begin the text buffer at address $0700. Be sure to hit return twice after all of the text is entered. This inserts $00 at the end of the text to signal the end of the text. It also stores the end text address in locations $00E1-00E2.

You may also want to employ one of the user-function keys to run the program. Program JMP 0400 at location $010C and JMP 0200 at location $010F. The program will begin running when you hit the F1 key. The F2 key starts the clock running and begins the voltage-monitoring section of the program. If the program is to be stored on tape, be sure to store addresses $010C-$0112 on the tape also. This allows the program to use F1 and F2 after being loaded from cassette tape.

The program in Listing 2 can be broken into four parts. Addresses $0200-0274 initialize the real-time clock and provide a printer and display for the clock. Locations $0300-033C contain the interrupt routine for the clock. The clock routine is similar to the one by Marvin L. DeJong in the March 1979 issue of *Micro*. The major change is in the clock display routine.

Locations $0400-04AD output the original prompt to the display and printer. They also allow you to input the time at which the main program will be started. Locations $0500-0593 monitor the voltage and output the change and time when the change occurred.

In the first section ($0200-0274), the first 18 instructions initialize the values for the 6522 VIA timer 1 interrupts. Locations $0215-021C store $C342 as the clock interrupt frequency. Locations $0226-0274 contain the display routine, which outputs the time to the display and printer. At location $0232 the X register is loaded with the value $0D. This is done because the value of the X register determines a character's position on the display. The actual display of the character is accomplished at location $024E by JSR E97A.

Location $E97A, an address in the AIM 65 monitor, is the start of a subroutine that takes an ASCII character in the accumulator and outputs it to the display and printer. The only problem is that the time is

Listing 2.

```
0200 78 SEI
0201 A9 LDA #00
0203 8D STA A404
0206 A9 LDA #03
0208 8D STA A405
020B A9 LDA #C0
020D 8D STA A00E
0210 A9 LDA #40
0212 8D STA A00B
0215 A9 LDA #42
0217 8D STA A006
021A A9 LDA #C3
021C 8D STA A005
021F A9 LDA #EC
0221 85 STA 00
0223 58 CLI
0224 00 BRK
0225 EA NOP
0226 A5 LDA 01
0228 85 STA 04
022A A5 LDA 02
022C 85 STA 05
022E A5 LDA 03
0230 85 STA 06
0232 A2 LDX #0D
0234 A0 LDY #06
0236 B9 LDA 0000,Y
0239 85 STA 10
023B 98 TYA
023C 85 STA 11
023E A0 LDY #04
0240 66 ROR 10
0242 88 DEY
0243 C0 CPY #00
0245 D0 BNE 0240
0247 A5 LDA 10
0249 29 AND #0F
024B 18 CLC
024C 69 ADC #30
024E 20 JSR E97A
0251 E8 INX
0252 A5 LDA 11
0254 A8 TAY
0255 B9 LDA 0000,Y
0258 29 AND #0F
025A 18 CLC
025B 69 ADC #30
025D 20 JSR E97A
0260 E8 INX
0261 88 DEY
0262 C0 CPY #03
0264 F0 BEQ 026E
0266 A9 LDA #3A
0268 20 JSR E97A
026B 4C JMP 0236
026E 20 JSR EA13
0271 20 JSR EA13
0274 4C JMP 0500
0277 EA NOP
0300 48 PHA
0301 E6 INC 00
0303 D0 BNE 0337
0305 F8 SED
0306 18 CLC
0307 A5 LDA 01
0309 69 ADC #01
030B 85 STA 01
030D C9 CMP #60
030F 90 BCC 0333
0311 A9 LDA #00
0313 85 STA 01
0315 18 CLC
0316 A5 LDA 02
0318 69 ADC #01
031A 85 STA 02
031C C9 CMP #60
031E 90 BCC 0333
0320 A9 LDA #00
0322 85 STA 02
0324 18 CLC
0325 A5 LDA 03
0327 69 ADC #01
0329 85 STA 03
032B C9 CMP #24
032D 90 BCC 0333
032F A9 LDA #00
0331 85 STA 03
0333 A9 LDA #EC
0335 85 STA 00
0337 AD LDA A004
033A D8 CLD
033B 68 PLA
033C 40 RTI
033D EA NOP
0400 A2 LDX #00
0402 A0 LDY #00
0404 A9 LDA #00
0406 8D STA 04F0
0409 B9 LDA 072E,Y
040C C9 CMP #0D
040E F0 BEQ 0418
0410 20 JSR E97A
0413 E8 INX
0414 C8 INY
0415 4C JMP 0409
0418 20 JSR EA13
041B A2 LDX #00
041D C8 INY
041E EE INC 04F0
0421 AD LDA 04F0
0424 C9 CMP #07
0426 D0 BNE 0409
0428 A0 LDY #00
042A B9 LDA 07AE,Y
042D C9 CMP #0D
042F F0 BEQ 0439
0431 20 JSR E97A
0434 E8 INX
0435 C8 INY
0436 4C JMP 042A
0439 20 JSR EA13
043C C8 INY
043D A2 LDX #00
043F EE INC 04F0
0442 AD LDA 04F0
0445 C9 CMP #0E
0447 D0 BNE 042A
0449 20 JSR E95F
044C 8D STA 04F1
044F 20 JSR E95F
0452 8D STA 04F2
0455 20 JSR E95F
0458 8D STA 04F3
045B 20 JSR E95F
045E 8D STA 04F4
0461 20 JSR E95F
0464 8D STA 04F5
0467 20 JSR E95F
046A 8D STA 04F6
046D A2 LDX #01
046F A0 LDY #04
0471 3E ROL 04F0,X
0474 88 DEY
0475 C0 CPY #00
0477 D0 BNE 0471
0479 BD LDA 04F0,X
047C 29 AND #F0
047E 18 CLC
047F 9D STA 04F0,X
0482 E8 INX
0483 E8 INX
0484 E0 CPX #07
0486 D0 BNE 046F
0488 A2 LDX #02
048A BD LDA 04F0,X
048D 29 AND #0F
048F 18 CLC
0490 CA DEX
0491 7D ADC 04F0,X
0494 9D STA 04F0,X
0497 E8 INX
0498 E8 INX
0499 E8 INX
049A E0 CPX #08
049C D0 BNE 048A
049E AD LDA 04F1
04A1 85 STA 03
04A3 AD LDA 04F3
04A6 85 STA 02
04A8 AD LDA 04F5
04AB 85 STA 01
04AD 60 RTS
0500 A9 LDA #FF
0502 8D STA A002
0505 A9 LDA #00
0507 8D STA A003
050A A2 LDX #00
050C A0 LDY #00
050E A9 LDA #02
0510 8D STA 0625
0513 8D STA A000
0516 AD LDA A00F
0519 9D STA 0600,X
051C DD CMP 0650,X
051F D0 BNE 0542
0521 BD LDA 0600,X
0524 9D STA 0650,X
0527 E8 INX
0528 CE DEC 0625
052B AD LDA 0625
052E C9 CMP #00
0530 D0 BNE 0513
0532 8D STA A000
0535 AD LDA A00F
0538 9D STA 0600,X
053B DD CMP 0650,X
053E D0 BNE 0546
0540 F0 BEQ 054D
0542 C8 INY
0543 4C JMP 0521
0546 C8 INY
0547 BD LDA 0600,X
054A 9D STA 0650,X
054D 98 TYA
054E C0 CPY #00
0550 F0 BEQ 050A
0552 A2 LDX #00
0554 BD LDA 0700,X
0557 C9 CMP #0D
0559 F0 BEQ 0562
055B 20 JSR E97A
055E E8 INX
055F 4C JMP 0554
0562 20 JSR EA13
0565 A2 LDX #00
0567 BD LDA 0650,X
056A 20 JSR E97A
056D E8 INX
056E E0 CPX #03
0570 D0 BNE 0567
0572 BD LDA 0713,X
0575 C9 CMP #0D
0577 F0 BEQ 0580
0579 20 JSR E97A
057C E8 INX
057D 4C JMP 0572
0580 20 JSR EA13
0583 A2 LDX #00
0585 BD LDA 0721,X
0588 C9 CMP #0D
058A F0 BEQ 0593
058C 20 JSR E97A
058F E8 INX
0590 4C JMP 0585
0593 4C JMP 0226
0596 EA NOP
```

stored in decimal form in the memory.

Before the characters can be displayed, they must be converted to the ASCII format. This is accomplished by first storing the two-digit number in an address where it can be manipulated. The number is rotated right four bits so that the most significant nibble (MSN) is now the least significant nibble (LSN). It is then ANDed with $0F to save only the LSN. $30 is then added to produce the correct ASCII code for any decimal number 0–9.

To display the second number of the pair, the original two digits are once again retrieved from memory. It is not necessary to rotate this data since the LSN is in the proper place. It is simply converted to ASCII in the same manner as the other and displayed. Between the hours, minutes and seconds, a colon is displayed.

After the time has been displayed, there is another jump to a monitor subroutine. The subroutine at $EA13 outputs a carriage return and line feed to the display and printer. This is done twice. The final instruction executes a jump to another section of the program, which looks for a change in voltage.

The second section, from locations $0300–033C, is the clock. This section is used every time an interrupt is generated by the timer in the VIA. When 24 hours are up, the clock resets itself and continues to run. The clock continues to keep fairly accurate time even while other parts of the program are being run.

The third section, locations $0400–04AD, outputs the original prompt and allows the user to input the time he wishes to start the program. The use of the text buffer as the source of the prompt is accomplished by using the X register as the display counter, the Y register as the text buffer location counter and $04F0 as a text line counter. The characters are loaded into the accumulator by using absolute indexed addressing.

The absolute address is the starting address of the text. The Y register is then added to the absolute address to obtain the correct address for the character to be displayed. The Y register is then incremented to obtain the consecutive characters. A carriage return (ASCII $0D) will be found at the end of each line.

When a CR is loaded into the accumulator, the program jumps to a subroutine that outputs a carriage return and line feed to the display and printer. The Y register is incremented to be ready to fetch the next character. The X register is set to zero in order that the next line of characters appears at the proper place on the display and printer. Address $04F0 is then incremented and tested to see if all the lines of the text have been output. If not, the program branches back to $042A, ready to fetch the next line. Otherwise, the program looks at the keyboard using another subroutine.

At this point, the user inputs the time he will start the program. The clock will use this time as its starting time. The hour, minute and second times are input as two-digit numbers (eg., 01:15:30). Since each number is in ASCII format as it is taken from the keyboard, the two numbers for each division must be converted to decimal form and combined into one byte. This is done at locations $046D–$049C.

The decimal numbers are then loaded into the proper addresses for use by the clock. The program then returns to the monitor. When the time input to the clock is reached, the user hits F2 and the program begins to run.

The final section ($0500–0599) reads data from the meter and outputs the changes to the display and printer. This is accomplished by using ports A and B of the 6522 VIA. As it is presently structured, the program will read three digits by outputting on port B which digit it wishes to look at. It then reads the data on port A. Both the digits read and those output are in BCD format.

The first four instructions initialize port B as an output port and port A as an input port. While this program initializes the entire port as one or the other, each individual bit can actually be selected as an input or output. The requested digit is output on port B by loading address $A000 with the requested digit.

```
<[>THIS PROGRAM WILL
MONITOR LINE VOLTAGE
AND RECORD THE
CHANGES. IT WILL
ALSO RECORD THE TIME
OF THE CHANGES.
  TO INITIALIZE THE
PROGRAM, ENTER THE
TIME IN THIS FASHION
      HR:MIN:SEC.
ALLOW TWO DIGITS PER
UNIT. THEN HIT F2 TO
START THE PROGRAM AT
THE RIGHT TIME.
112220
<]>11:22:20

THE VOLTAGE IS
042VRMS-60HZ.
THE TIME IS 11:22:22

THE VOLTAGE IS
046VRMS-60HZ.
THE TIME IS 11:22:31

THE VOLTAGE IS
074VRMS-60HZ.
THE TIME IS 11:22:33

THE VOLTAGE IS
120VRMS-60HZ.
THE TIME IS 11:22:36

THE VOLTAGE IS
114VRMS-60HZ.
THE TIME IS 11:22:38

THE VOLTAGE IS
117VRMS-60HZ.
THE TIME IS 11:22:40

THE VOLTAGE IS
116VRMS-60HZ.
THE TIME IS 11:22:42

THE VOLTAGE IS
117VRMS-60HZ.
THE TIME IS 11:22:44
```

Listing 3. Sample run.

The most significant digit is chosen first. It is then compared with the previous reading. If it is not the same, the reading is stored and the Y register is incremented.

After the three digits have been read, the Y register is checked to see if any digits have changed. If not, the process is started again. If the data has changed, more text is taken from the text buffer. It explains what the data is. The new voltage is then output to the display and printer. The clock is then checked, and the current time is also output. The process then begins again.

The simplicity of using the I/O ports should now be evident. All you have to do is to select the function of the port by loading the proper address with 1s or 0s. Even the individual bits can be independently programmed for function. The port can then be read by loading the accumulator with the value found at the port's address. Data can be output on a port by storing the data at the port's address. What could be simpler than that?

This program looks at a different digit approximately 8000 times per second. This means a complete reading is done 44 times per line cycle. The only drawback is that most digital voltmeters do not update their ouput nearly that fast. Your ability to use this program to record voltage fluctuations or line spikes will be limited by the meter used for the measurements.

```
0000 A9 LDA #00
0002 85 STA A0
0004 85 STA A2
0006 A9 LDA #02
0008 85 STA A1
000A A0 LDY #00
000C B1 LDA (A0),Y
000E C9 CMP #0D
0010 F0 BEQ 002F
0012 20 JSR E97A
0015 C0 CPY #FF
0017 F0 BEQ 001D
0019 C8 INY
001A 4C JMP 000C
001D 18 CLC
001E A5 LDA A0
0020 69 ADC #FF
0022 85 STA A0
0024 A5 LDA A1
0026 69 ADC #00
0028 85 STA A1
002A C8 INY
002B C8 INY
002C 4C JMP 000C
002F 20 JSR EA13
0032 E6 INC A2
0034 A5 LDA A2
0036 C9 CMP #10
0038 F0 BEQ 0042
003A C0 CPY #FF
003C F0 BEQ 001D
003E C8 INY
003F 4C JMP 000C
0042 EA NOP
0043 60 RTS
0044 EA NOP
```

Listing 4. Alternative method of text display.

Listing 3 is a sample run of the program. Note that when inputting the time, you do not separate the hours, minutes and seconds by colons. These are inserted by the program for the output. Although this program was run on an AIM 65 with 4K RAM, it is easily run in the 1K version. The only change involved is the reassigning of addresses.

Alternative Method

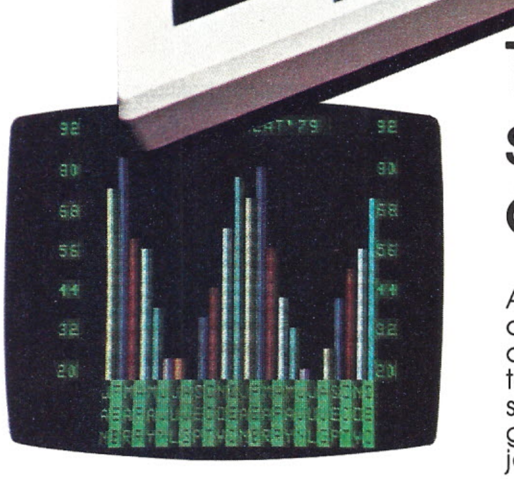
This program uses absolute indexed addressing to output text from the text buffer. The advantage of this method is that it takes few steps to accomplish. The drawback is that it can only make use of text no longer than 256 characters.

Listing 4 shows an alternative method of outputting text. It uses indirect indexed addressing. While it takes more steps to accomplish the same goal, it can address text of up to 64K characters in length.

This particular example starts the text address at $0200. This information is stored at locations $A0 and $A1. Address $A2 is used as the text line counter.

Once again it is the carriage return (ASCII $0D) that is used to detect the end of a line. The Y register is incremented to $FF and then added to the address found at $00A0 and $00A1. $00A0 contains the low-order byte of the text address, while $00A1 contains the high-order byte. Since the address at $A0 and $A1 can be changed, any length text at any location can be addressed. This method has been used successfully to print text over 1K in length.

This program uses many of the unique features of the AIM 65. The text editor, parallel I/O ports and the wide variety of monitor subroutines provide a combination of features not to be found anywhere else in the world of single-board computers. The AIM is easy to use once the user has unlocked a few of its secrets and is able to make use of the monitor subroutines and other features. I hope that this program has given you some new insights into programming the AIM 65. ■



The only computer with color, sound, user programmability and expandability at $599.

The Imagination Machine offers more at its price than any other personal computer on the market today.

Consider these features: 9K RAM, with 14K BASIC in ROM,53-key typewriter keyboard. A fine resolution picture, generated on your television set or monitor in 8 colors!

A built-in, dual-track cassette tape deck with 1500 baud rate, for APF's digitally recorded, "saturated," tape programs. A built-in sound synthesizer. And two, built-in, game style controllers, with joysticks and numeric keypads.

When you want to go beyond APF's library of educational, home-and-personal management or entertainment programs... when you want to create your own programs... you can. The Imagination Machine is programmable in BASIC and 6800 machine language. The Imagination Machine is also expandable. Just add our "Building Block", an optional, four-port expansion device, and you can hook up a printer, telephone modem, and additional memory cartridge or mini-floppy disk drive.

For the name of your nearest Imagination Machine dealer call, TOLL FREE: 1-800-223-1264. (New York residents call: (212) 758-7550) or write: APF Electronics, Inc. 444 Madison Avenue, N.Y., N.Y. 10022.

$599. Manufacturer's suggested retail price.

APF electronics inc.

"YOUR LIFE WILL NEVER BE THE SAME."

Physician: Automate Thyself

Doctors who have computerized their accounts are saving both time and money.

Robert Charles Bauman
509 Wolcott Rd.
Wolcott CT 06716

Why would an optometrist in practice less than 12 months need a computer system in his office? If your background in computers is limited, you might imagine that a computer system — even a very small one — would only be necessary to process thousands of bills or perhaps prepare payroll checks for hundreds of employees. After the year we have been in practice, our office does not process thousands of patient bills, nor do we have hundreds of employees. But our computer is busy every day.

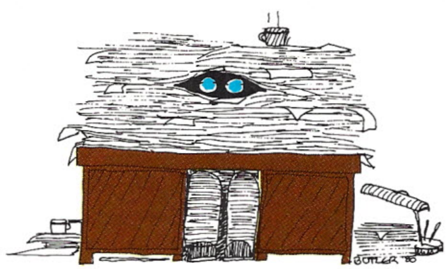
Eye-opening Applications

Our microcomputer is a versatile communication tool. It is an enthusiastic spokesman for my office and optometry in general. It tells my patients that this office is unique, that we really care about our patients as individuals.

A microcomputer is a useful addition to your office when it becomes too time consuming or expensive for your staff to send a personalized thank-you letter to all patients seen in your office. The computer can also send a personal recall letter to each patient due for recheck. The few monthly bills that our office sends out to patients are easily handled by office staff. But in most offices, the staff cannot send a personal thank-you letter to all patients seen, or a personalized recall letter to all patients.

With recent advances in the electronics field, a computer system that cost $50,000–$60,000 ten years ago now costs less than $10,000. This revolution in electronics can put the power of the computer within reach of most practitioners. Studies estimate that 80 percent of all microcomputers purchased will be used to perform word-processing tasks. Manipulation of the written word can allow you to compose and edit letters, manuscripts and reports, for example. This article was composed, corrected and edited on my system. With such systems, a poor typist such as myself can still generate good-looking text at speeds that approach or exceed those of a typist with expert skills.

Telling our patients what we do is as important as the services we actually perform. Large corporations have made effective use of mass communication. Now optometrists have the same capability.



A microcomputer is useful in your office when it becomes too expensive to send personalized letters to all patients.

In all but the largest optometric offices, a computer does not have to store patient records; the file cabinet can handle that. In most offices, the computer will not keep frame inventories, nor will it play chess. Although a computer is capable of performing these tasks, our computer is put to other uses.

In our office the computer will generate a thank-you letter to each patient. In this letter a recall date of so many months is mentioned. This establishes in the patient's mind the need for regular eye care. The date of the last visit, as well as any other informative information, is mentioned. With high postage costs today it is not cost-effective to use a postcard for patient recall notices. This type of notification system tells your patients that this doctor is the same as all the others. In his office, patients are just numbers.

For a few cents more, a computer can generate a personal note that shouts, "My doctor really cares!" Such letters not only can improve recall performance but can actually result in more referrals. Today, computer technology can generate a letter and address the envelope in less than 40 seconds. Such letters are indistinguishable from letters typed by hand, except the computer never presses the wrong keys.

With a little more imagination, this communication device can be used to even greater advantage. Do you think your contact lens patients or prospective contact lens patients would respond to a personal letter informing them of a new development in the contact lens field? Those practitioners involved in visual training generate scores of reports. Why not let the computer communicate for you? Teachers, school nurses and parents will all respond in a positive manner to a personal report from your office.

As professionals, we have only a limited supply of time to share with our patients. Automation, if properly used, can free the doctor and staff from busywork so more time can be spent on tasks that take judgment and insight.

Purchasing a Computer System

Equipment purchased must justify its cost by enhancing the practice and removing an element of drudgery for doctor or staff. A microcomputer excels in both areas. Computers, unlike people, are at their best when performing repetitious

tasks. Any activity that is performed on a regular basis can be better handled by the computer.

To make the purchase and integration of a computer system in your office as painless as possible, do some homework. Learn what you can reasonably expect a computer system to do. Most areas of the country are within a few hours' drive of a retail computer store that specializes in systems for the professional practice or the small bussinessman.

These stores generally have abundant reference materials on hand. Computer journals that are entirely devoted to the small computer system are available. These magazines provide an excellent education; you may find the advertisements useful. Easy-to-understand books on the subject of microcomputers are informative. Computer clubs are generally willing to help the beginner avoid costly errors.

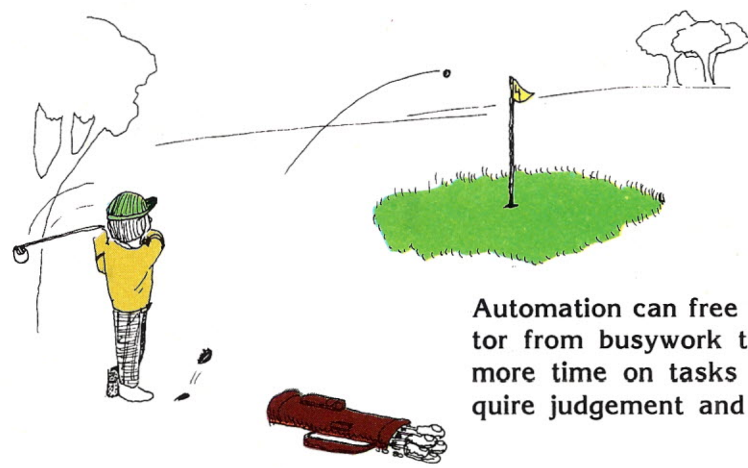
A few general principles will make the purchase easier. A small computer system does not cost $1000. Although hobby computers are available in that price range, a useful system will range in cost from $4000–$10,000 for hardware. Although the cost of computers may fluctuate slightly in the future, significant price reductions are unlikely. Those waiting for a $10,000 computer system to decrease in cost to $4.95 will still be waiting five years from now.

Above all, I have learned that all computer equipment will need service eventually. If cost is a major concern, much of the necessary equipment can be purchased in kit form, with significant savings. A detailed knowledge of electronics is not necessary. I built my system in kit form at a total cost of $4500. Although I had built electronic kits before, my experience in electronics was minimal. If time is an overriding factor, then purchase assembled equipment.

What You Need

All computer systems have some components in common. Our system's basic computer is a small box about the size of a stereo receiver. The cabinet holds the memory circuits and controls all other components of the system. Simple systems have fewer memory circuits; powerful systems have more. Many computers, such as mine, have made provision for adding additional memory circuit boards as the need arises. My system has the ability to store 32,000 (32K) pieces of information, or bytes. A capacity of 24,000 (24K) bytes is about minimal.

In addition to the computer, we must have equipment to communicate our desires to the computer. Most often this is accomplished by a keyboard terminal at-



Automation can free the doctor from busywork to spend more time on tasks that require judgement and insight.

tached to a modified television screen.

The television-type terminal allows me to compose a report, for example, and make any changes or corrections on the screen. Once satisfied, I can instruct the computer to type the final version with 100 percent accuracy. This greatly speeds up the process of writing letters and manuscripts. In addition, the letter can be filed in the computer's memory and recalled when desired.

Although the computer can communicate with me by using the television-type terminal, a useful system will have a device that the computer can control to generate written copy, or hard copy. Although many printers are available, the daisy-wheel type is best. This kind of printer is fast—about 55 characters per second can be printed. The output from this type of printer cannot be distinguished from print generated on the best electronic typewriter. Many computer printers can also function as a standard typewriter when not connected to the computer.

In addition to the memory circuits contained in the computer itself, provision has to be made for permanent external storage of information. This is necessary because most computers "forget" what is in their memory when the power is turned off. External storage devices retain memory even after the power is turned off.

Most often magnetic disks are used to store this information. These disks resemble 45 rpm records. All programs and data files are stored on such disks, which are then placed in a disk drive unit to be read by the computer. This type of device also allows for easy duplication of data disks that store valuable information, so a duplicate disk can be made.

Software Requirements

Up to this point I have mentioned the equipment or hardware necessary for a complete computer system. In order for any computer system to perform useful work, it must be told exactly what to do. Hardware is not useful until software programs that tell the computer to do exactly what is desired are acquired.

Most programs written for microcomputers use a simple computer language called BASIC. Many first-time computer users have taught themselves this programming language. Self-study manuals are available on the subject. Even though simple programs can be written by the novice, more elaborate computer programs such as those controlling computer billings require professional programming assistance.

Software availability is important. More than one doctor has purchased a computer system only to find that no software exists to perform the tasks he wants done. Custom software is available, but quality software is expensive. Since software from one system may not run on another system, computer equipment must be purchased from reliable sources. Believe only what you have actually seen demonstrated before you make any equipment purchases.

Good software allows even the inexperienced to operate a computer properly. It leads the operator "by the nose" so errors are minimal. Although I have written all software myself, busy practitioners will want to rely on purchased software.

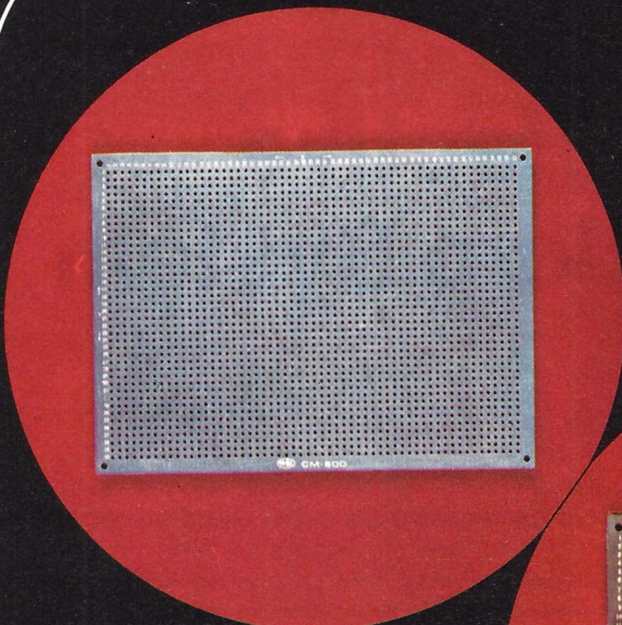
Conclusion

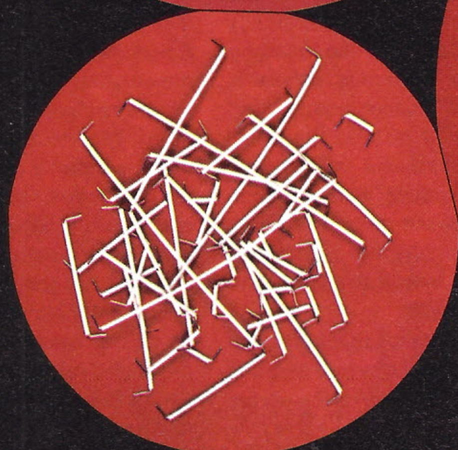
Although a computer system is more difficult to purchase than many other types of office equipment, the doctor who takes the time to educate himself on the subject or can obtain help from an individual familiar with microcomputer systems will find the experience painless.

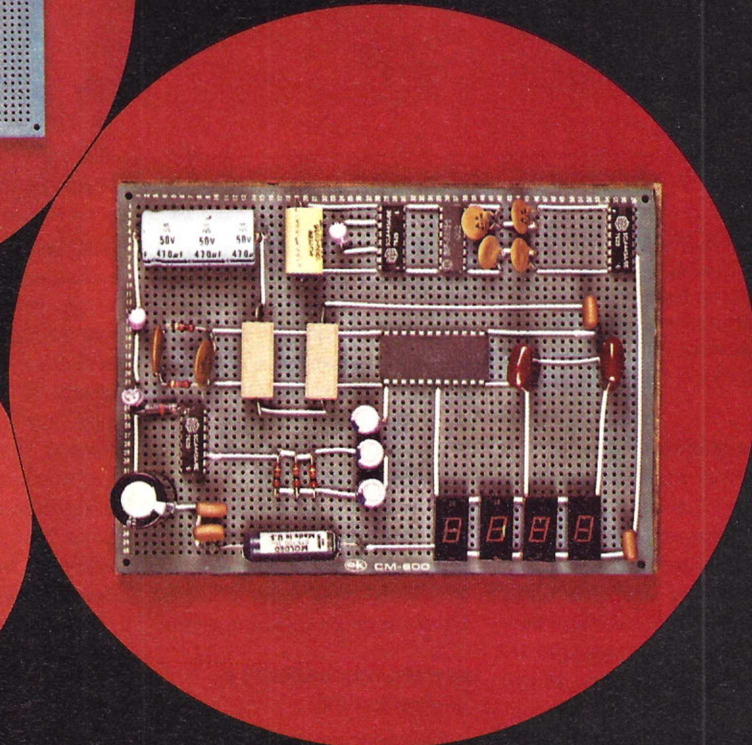
If properly used, the office computer can convert satisfied patients into enthusiastic patients. It can free the doctor and staff from many chores. Put this tool to work in your office. ■

ok®

CM-600 Circuit Mount







CM-600 $6.95*
RW-50 $2.98*

NEW CM-600 SOLDERLESS PROTOTYPE BOARD

CM-600 is a unique system for solderless construction of circuit prototypes, useful to both engineers and hobbyists. The CM-600 is a neoprene board 4½″ (114mm) x 6″ (152mm) with 2280 holes on .100″ (2.54mm) centers. Standard components including DIP's are mounted by simply inserting leads into the holes in the long life neoprene material. Interconnections are easily made using 20 or 22 AWG (0,8 or 0,65mm) wire jumpers. Positive contact is assured by the elasticity of the hole, which compresses the leads together. To remove components or leads, simply pull out. This facilitates easy circuit changes making it ideal for breadboarding experimental circuits. CM-600 also features numbered rows and columns for easy reference.

Accessory Kit RW-50 contains 50 pcs of AWG 20 (0,8mm) insulated jumper wires of assorted lengths from ½″ (13mm) to 4″ (100mm). Both ends are stripped and bent 90° for easy insertion. In stock directly from

OK Machine & Tool Corporation ✓21
3455 Conner St., Bronx, N.Y. 10475 U.S.A.
Tel. (212) 994-6600 Telex 125091

*Minimum billings $25.00, add shipping charge $2.00
New York State residents add applicable tax

✓Reader Service—see page 258

A Cost-Effective Multi-user Micro System

When a 16-station system was needed, this clinic developed its own—and saved a bundle.

Wayne M. Kashinsky
Dr. Raymond G. Romanczyk
Dept. of Psychology
State University of New York
at Binghamton
Binghamton NY 13901

In educational or clinical settings, it is often desirable to have quick access to a file containing information about the progress or treatment of an individual. There are also other settings that require many people to have simultaneous access to pre-stored data. In the past, these applications required large, time-sharing computer systems, which were often prohibitively expensive and complex. Furthermore, the addition of each user terminal is a significant expense (approximately $1000).

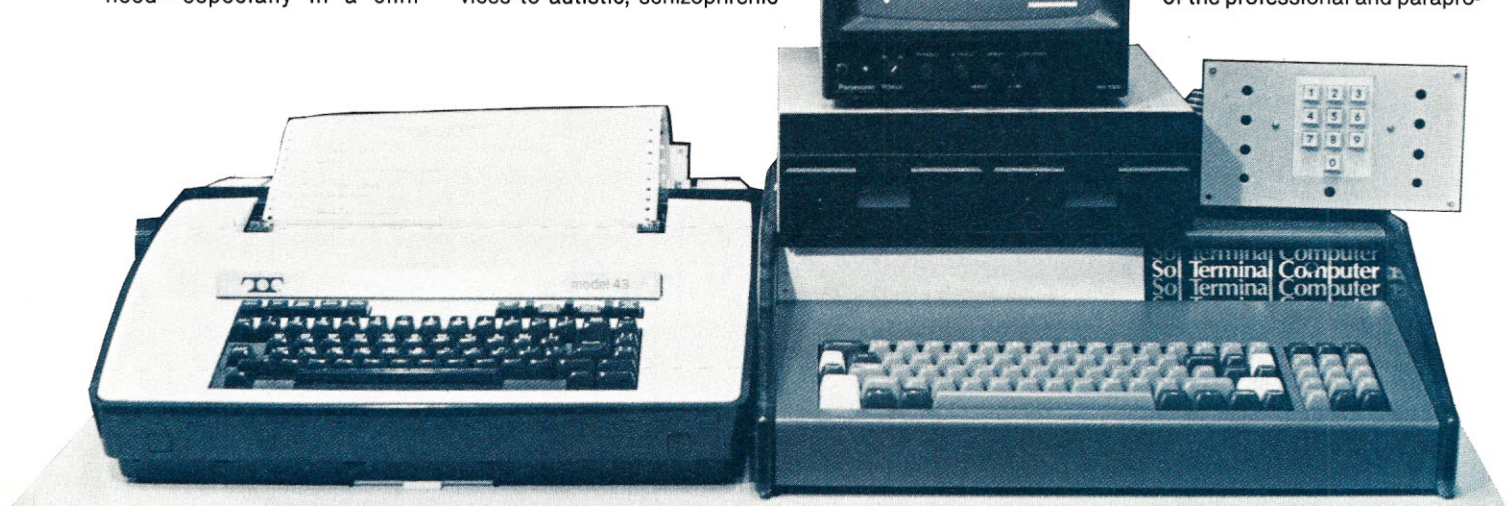
To meet this multi-user need—especially in a clinical/educational setting—we developed a low-cost, flexible, 16-station system. The cost of the complete computer system, including a printer, is less than $5000, with each user station costing approximately $100. Thus, if you can keep the amount of data needed on-line for immediate access on two to four floppy diskettes (under 2 megabytes), you can make use of our implementation of a multi-user microprocessor system.

Specific Application in a Clinical Setting

The site of our multi-user system is the Children's Unit for Treatment and Evaluation, an intensive treatment facility for severely disturbed children. It is located within the Department of Psychology at SUNY-Binghamton. The unit provides services to autistic, schizophrenic and severely emotionally disturbed children. At a descriptive level, the children typically manifest some of the following problems: muteness, echolalia, psychotic speech, gaze aversion, social isolation (autism), temper tantrums, aggression, self-destructive behavior and inappropriate social interactions. Typically, the children are also deficient in academic and self-help skills.

The goal of the unit's program is to provide intensive services to help the child return to his school system and participate in the special-education facilities available. Thus, the unit functions as the first stage of remediation to prepare the child for the special-education classroom. Therefore, the treatment priorities stress the elimination of aggressive, self-destructive and tantrum behavior patterns and the acquisition of speech and basic language skills and of social, self-help and play skills, as well as readiness skills (i.e., prolonged attention, following simple directions, being able to work in both a group setting and independently and simple discriminations).

The treatment of such children has historically been a difficult problem. The unit has been able to amass substantial resources to confront these disorders. As an example, the size of the professional and parapro-



fessional staff is such that there are as many staff members as children. Thus, much of the therapeutic and educational interactions are conducted on an individualized, one-to-one basis.

Such a program requires extensive collaboration among all staff members to ensure consistency, and the amount of information collected each day on each child for each specific therapeutic and instructional program is extensive. Indeed, it is not unusual to record behavior episodes on a minute-by-minute basis and learning acquisition information on an individual response basis. Thus, several hundred pieces of information are collected on each child each day.

In order to improve the efficiency of the staff, provide them with accurate, properly analyzed and current information and to relieve them of certain aspects of routine data analysis and administrative burdens, we implemented a multi-user microcomputer system. Since the physical plant of the unit is extensive—encompassing approximately 22 rooms—it was essential that the per terminal cost be kept low. It was also critical that the system be easy to use, given the number of people who would be utilizing the system.

Computer Functions

Any staff member may interrogate the system concerning the current status of any child, unless selected, confidential aspects of the file are locked for controlled access by senior staff. Information on-line for each child includes schedule information, program history, medical information and bar graph displays of specific data sets. Each data set in our specific implementation may contain up to 200 data points as well as relevant text information. Scaling of data for graph display is done automatically or may be overridden manually at the terminal. By using data compression techniques and byte-level addressing of disk files, you can store over 300 such data sets on-line on the system.

Program history contains a description of each clinical and educational treatment program, its start date, end date, degree of success and status (active or terminated). Schedule information contains a breakdown of scheduled activities for each child by the half hour for each day of the week. Medical information specifies important data, such as specific allergies, medication type and dosage and dietary restrictions, needed on a day-by-day basis.

The system also monitors time of day, operating a prompt system for schedule changes, indicating when activities should start and stop, typically on the half hour. In addition, the system can prompt the staff at special times during the day, such as when medication is to be dispensed or other time-dependent actions are to take place. In instances of prompted functions, the terminal utilizes a visual display of information plus an audio signal that associates a specific tone with each function, so that staff need not be initially visually oriented to the terminal in order to be made aware a function is engaged.

In the event of an emergency, a staff member can press the number 9 on the keypad to summon the instant response of support and supervisory staff. This engages a warble tone at all terminals and displays a floor plan of the unit on the screen and a flashing asterisk (*), indicating the source (see Example 1). The alarm will continue until it is reset from that room by an assisting staff member. The system logs the time and location of the incident on a printer.

In addition, other specific events can be automatically logged and printed, specifying the time, location, child involved and specific behavior displayed.

This data-logging occurs when a staff member indicates a selection—via keypad—from the menus of children and behavior that appear on the screen in specific therapy rooms. The location and child information is stored on-line, so that each room has a unique menu reflecting the specific children present.

```
condition : RED      emergency in :ACADEMIC CLASSROOM
  +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  |     |     |     |     |     |     |     |     |     |     |
  |     |     |     |-----+-----|     | ----|     |     |---- |
  |-----+-----|     |           |     |     |     |     |     |
  |     |     |     |           |     | ----|     |     |---- |
  |     |     |     |     *     |     |     |     |     |     |
  |     |     |     |           |     | ----|     |     |---- |
  |     |     |     |           |     |     |     |     |     |
  |     |     |     |           |     | ----|     |     |---- |
  |     |     |     |           |     |     |     |     |     |
  +-----+-----+-----+-----------+-----+-----+-----+-----+-----+
    344   346   348   350   352   354   356   358   360   362
   ********* SUPERVISORY STAFF RESPOND IMMEDIATELY *********
```

Example 1. Emergency display.

```
          MESSAGE FOR STAFF - - - PLEASE RESPOND

               TIME IS -  1: 33

     **************************************************

          Dr. Romanczyk to the PLAYROOM

This request is :IMPORTANT

     **************************************************

1= ON MY WAY   2= COMING SOON   3=BUSY   4= NOT HERE

*********** REPLY IS **********
ON MY WAY
```

Example 2. Intercom/paging display.

Since there is a large number of staff members and a visual and auditory cuing system is the best method of locating a staff member and routing him to the proper room, an intercom/paging function was included (see Example 2). The system displays a menu of names and possible messages (e.g., "urgent," "telephone call," "at convenience"). After the person summoned makes a selection, a tone is transmitted to all terminals and the visual message is displayed until a response is entered, indicating the availability status of the requested person. All responses are visible at all terminals so that alternate staff can be alerted to the request.

Hardware

The major components of the system include an S-100 computer with disks, such as North Star, Cromemco or Processor Technology. We utilized a Sol computer with two dual-density North Star 5 inch disk units and a Model 43 Teletype (see Photo 1). Because the system was designed for ease of implementation and fast modification, we chose an interactive BASIC for programming. Although it takes milliseconds to process each program statement, this does not degrade the response time of the system. While BASIC language programming uses additional memory for overhead, requiring almost 40K for the complete package, its flexibility and ease of quick modification more than compensate for these shortcomings.

For our purposes, we used a simple, resettable 16-bit counter with a free-running .1 or 1 second time base as a real-time clock providing time of day and interval timing capabilities. Except for the serial ports, used by the programming console/printer, all keyboard data from the 16 stations is handled by two 8-bit parallel input ports and two 8-bit parallel output ports. Video and audio switching are accomplished by latched output ports directly driving 5 V reed relays.

The keyboards use a common bus line, so that only one station may interrogate the system at a

```
    condition : GREEN           MONDAY      5,28,79
     .............................................
Program change in effect for Bryon,Tara, and
  Aaron ----- Check with your supervisor and the
  program change log.

Staff meeting will be delayed 1/2 hour this
  week ONLY. Prepare for discussion of IEP
  progress update.

  Russ will update histograms at 3:00 today.

     .............................................
>>>>>>> My status is : CLEAR     I am available for your use
```

Example 3. General message display.

time. A single-gated tone generator/amplifier is used to output a beep whenever any key is pushed. This not only provides keypress feedback, but also alerts other stations that the computer is in use. An output bit is used to switch a resistor in the tone generator changing the frequency, and a second bit gates the 1 second time base into the tone generator, thus automatically producing a series of pulses without requiring software timing. These bits allow the tone system to produce many types of audio feedback, each used for a different auditory prompt function.

We implemented each station (terminal) with a low-cost touch-tone-like keypad for input and a TV monitor linked to a 64-character-per-line-by-16-lines memory-mapped video system in the computer. We modified a Zenith J-121 TV by cutting a wire connecting the if strip and the first video amplifier and then terminating the video amplifier input on the rear of the case.

This TV sells for approximately $90 and uses a power transformer for isolation. The keypad is a non-encoded Automatic Electric unit selling for under $10. The memory-mapped display is fast, filling the screen with information in less than one second (approximately 2000 cps). In addition, it allows high-speed, limited graphic displays by filling specific memory locations and makes updating of special fields easy.

Up to 16 keypads can be connected to provide one-out-of-10 code and a corresponding data available pulse, each feeding a common bus line, daisy-chained between the stations. Each keypad has ten DPST buttons wired as shown in Fig. 1. When a button is pressed, one pole is used to ground the appropriate bus line, which is held at +5 V through a 100 Ohm resistor on the interface card. The other pole is wired in common with all buttons on the keypad so that if any button is pressed, a data available pulse will be generated.

This bus structure has proven to be completely noise immune, even though we have used over 1000 feet of cable in our installation. Previous attempts at a bus using Tri-state logic did not prove acceptable.

The data available line from each station is terminated in a 16-line-to-binary converter providing the 4-bit address of the keypad. These bits, along with the 4-bit number of the key provided by the bus interface using the 74147 decoder IC, are input to an 8-bit port. Thus, on a keypress, a simple read and decode operation will provide both the station location and the digit. We used a software loop that requires two successive inputs of the same value to further reduce the possibility of noise.

Two memory-mapped video boards are connected to SPDT reed relays so that the output of either board can be switched to the monitors at any one station or group of stations. Under stan-

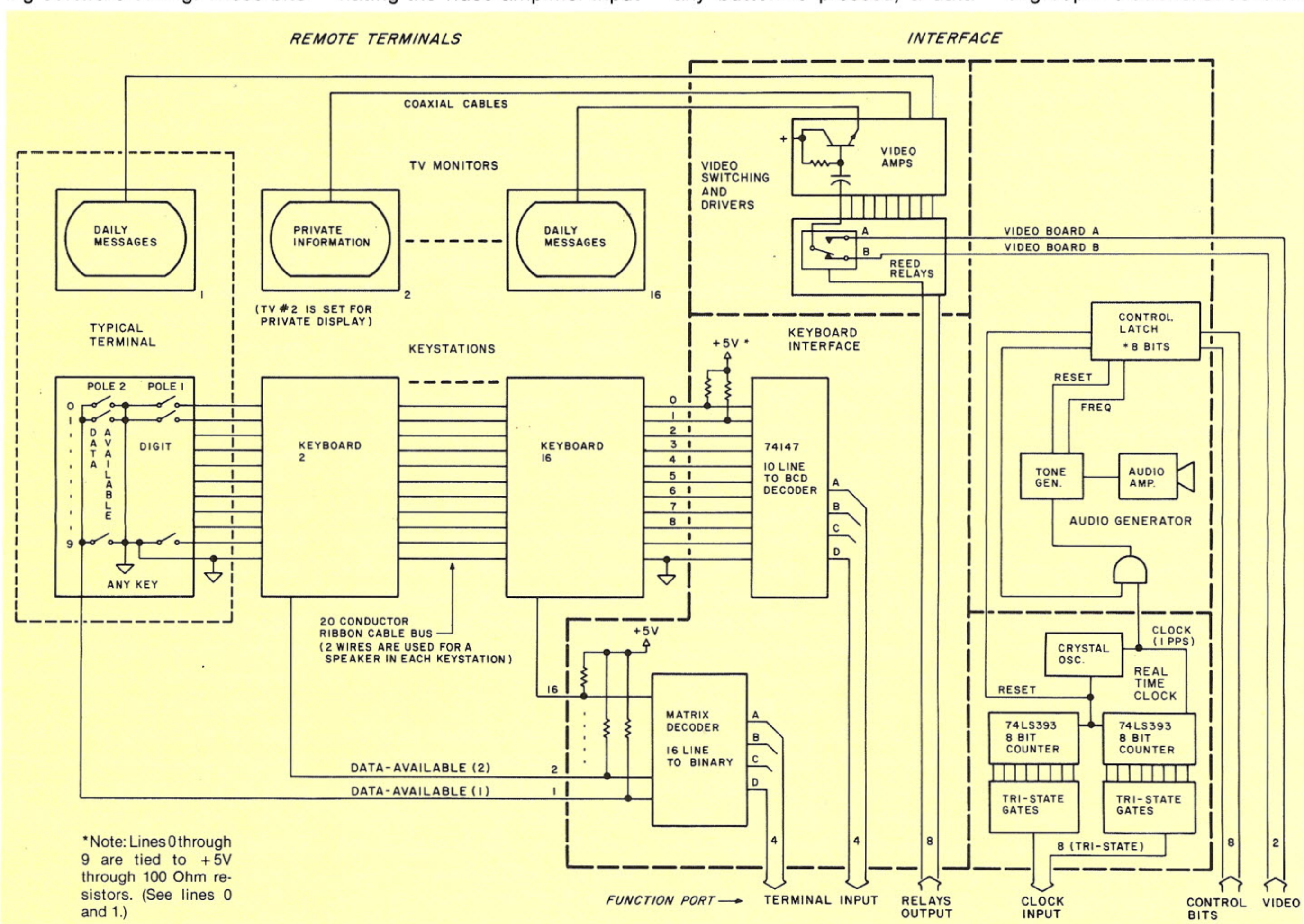
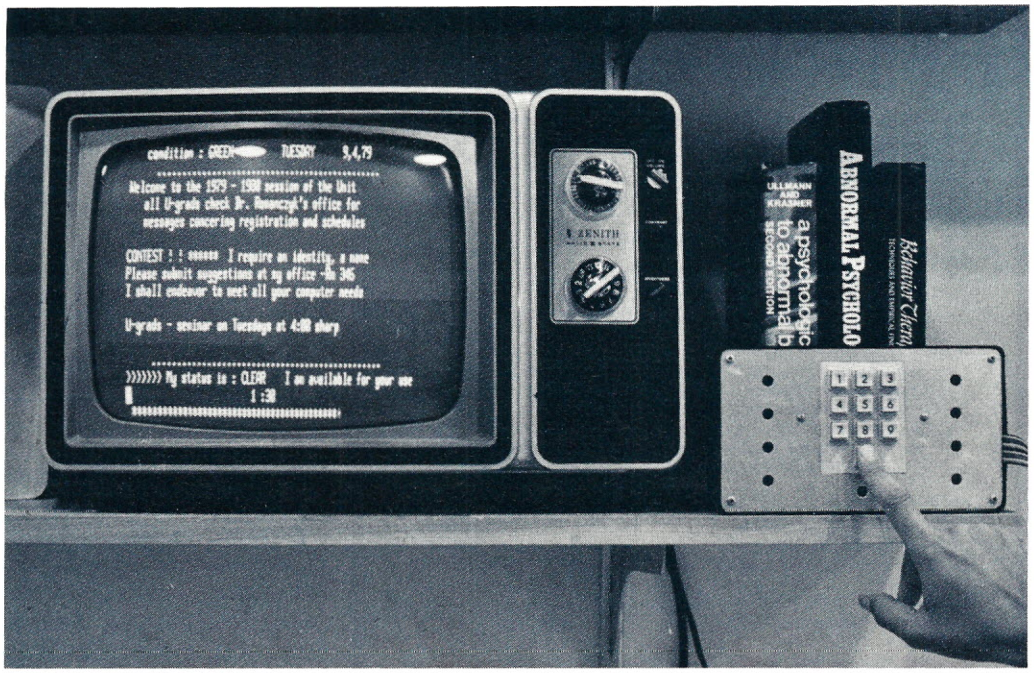


Fig. 1. Remote terminals interface.



TV monitor and keypad.

dard operating conditions, the normally closed contacts connect video board #1 to all monitors. Because the cable length between the computer and the monitors is long, the common pole of each relay drives a simple one transistor video amplifier, isolating each terminal and reducing interference that sometimes occurs when many monitors are connected in parallel.

If text is loaded into video board #2 and a relay or group of relays is closed, those monitors will display that information. This allows one board to be used for displaying general messages, systems status and time-dependent information, while the second board can be switched to provide private information to a requesting station.

While this method only allows one private display, the addition of extra video boards and relays would provide simultaneous access to special data by more than one station. Each keypad can be equipped with a multi-digit latched numeric display and an internal tone generator/speaker, making it ideal for such uses as timing and status reporting independent of the video system.

Software Organization

Once the program is started, all ports are initialized, the clock is set, and general messages are displayed on video board #1. Changes in the schedules and prompts are entered, and the program waits for a terminal request. During this wait time, many internal functions are being run. The clock is read and real time is updated on the monitors by directly writing into the memory-mapped display RAM. An array of special times is checked for a match, signaling a prompt display. A series of moving asterisks (*) is written on the bottom of the screen to indicate that the system is operational. During wait times, you can activate two special functions that allow recording of event data and printing of reports using a spooler.

When a button press is decoded, a menu for that function is displayed. The type of function determines if a private display is necessary or if all terminals should see the information. As an example, if button 0 is pressed, a menu of names and corresponding two-digit numbers appears on the screen. The system awaits a two-button keypress to define the child file that is to be interrogated and then opens his file.

A secondary menu appears on the screen providing a selection of the displays available. Another button press displays the requested information. During this time the clock is still read, and a higher priority function can override the current display. For example, a 9 button (emergency) will override any display.

Administrative Functions

In addition to the real-time operation of the computer system, it is also responsible for the preparation of budget projections, expense analyses and inventory control. Furthermore, the staff utilizes the computer system to prepare weekly child progress reports that can be forwarded to parents and relevant authorities. The software enables the user, by a single command, to search all child files and automatically format and print a report specifying all elements of the current schedule, program history and medical status as described above. It is also utilized as an extensive index system for cataloging professional literature as it relates to treatment issues and basic research, as well as a word processor for the preparation of reports and manuscripts.

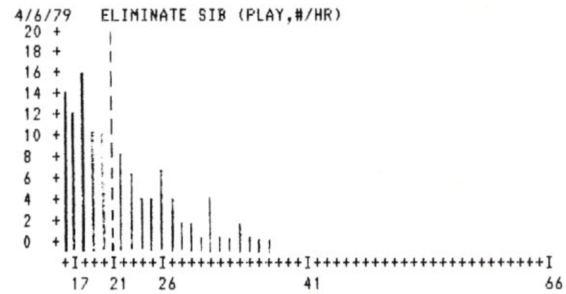
Additionally, the system is also used for complex analyses of child behavior. It is possible for staff members to enter observational data in 10-second bit format and have the computer provide them with a statistical analysis, such as conditional probabilities, phi coefficients or interrelationships among the discrete behavior patterns and overall frequency and distribution.

Perhaps one of the most important uses occurs at weekly staff meetings. Each child may be reviewed, with instant access to all relevant information, especially the bar graph displays, in much more depth and with greater speed than is typically possible. A large 25-inch monitor is present in the staff room, allowing all staff members to simultaneously view the information being presented.

In the past, review of all children would have to be spread over five to six weeks due to preparation time and the cumbersome aspects of pre-

```
PLEASE INDICATE YOUR CHOICE - - 1=ACTIVE     0=TERMINATED
                                5= ALL PROGRAMS


10/28/78 ELIMINATE AGGRESSIVE BEHAVIOR                              1
2/1/79 IMPROVING ARTICULATION (FOOD ITEMS)                          1
9/28/78 LANGUAGE AND READING SYNTAX - REBUS SERIES                  1
9/27/78 COLOR ID WITH PRONOUNS                                      1
3/26/79 DEVELOPING SPONTANEOUS SPEECH (4 WORD PHRASES)              1
9/25/78 ELIMINATE DISRUPTIVE BEHAVIOR (INDIVIDUAL LANG)             1
9/25/78 ELIMINATE AGGRESSIVE BEHAVIOR (INDIVIDUAL LANG)             1
2/29/79 INDEPENDENT ATTENTIVE SKILLS (STUFFING ENVELOPES)           1
2/28/79 DESCRIPTIVE SPEECH (PRONOUNS)                               1
3/26/79 SIGHT VOCAB (RESPONDING TO FUNCTIONAL SIGNS)                1
****** Press 0 to continue
3/28/79 COOPERATIVE PLAY (GAMES AND EXERCISES)                      1
3/28/79 ELIMINATE SIB                                               1
***** PRESS 7 TO CLEAR
```

Example 4. Sample of child's programs.



Example 5. Sample histogram.

senting this type of information. Thus, we are now able to review the children much more frequently than in the past, enabling us to provide greatly improved service delivery. The system is also useful in providing parents with up-to-date information at a moment's notice during visits to the unit.

The system has now been operational for two years, and there have been no failures whatsoever. The system as proved easy to use by the staff, which has responded enthusiastically to its implementation. Indeed, it requires only 10–15 minutes of instruction for a staff member to use the basic system, with perhaps an additional ½ hour to master all the off-line functions. The system has greatly enhanced the operation of the unit and is now an integral part of the organizational and therapeutic structure.

We wish to thank Micro World Computer Store in Johnson City NY for their assistance in the design of the hardware and software used in this system. ■

Attention

BARGAIN HUNTERS

Receive Hundreds of Classified Ads Like These Every Month

HARD DISK DRIVE Diablo Mod 31 1.2 MByte std. density. Includes power supp. and cable, rack mount slides, amd manual. Excellent condition. $450. Ca... 1601

IMPACT PRINTER 165 CPS Serial and parallel interfaces-Eight Selectable character sizes-Single and double width characters-uses standard plain paper - same mechanism as the integral data ... 1 year old $589, syst... Sta... St.,... 250

HEATHKIT H-11/DEC LSI-11 system, 32K Byte storage, reader 1 punch, video terminal, complete software. Cost $4500 assembled, $3500 kit. Like new. Sell for $2250. 305-962-6677. 2058 Griffin Rd., Ft. Lauderdale, FL 33312.

FOR SALE: Interdata (Perkin-Elmer) 7/16 Mini with 32KB core, front panel, 50A PWR supply. Includes HS tape reader, interfaces for LP, 2 (TTY), and RS-232 (Full duplex, programmable). Includes manuals and much SW (Basic, Fortran, OS, etc.,). $800 - After 6 P... 2035

COMPUTER AUTOMATION ALPHA 16; 16 k-word core memory,RTC, PF-R. Modified Mod. ASR-33 TTY Manuals, utilities, assemblers and many option boards - 16 bit I/O Driver, 16 bit I/O, Asynch modem contr. 64 bit output, 10 bit A/D - D/A. Fairly complete documentation. Up and running in Fortran. Not much more than TTY at $1000. **Herb Sauer, 303-494-8724.**

FOR SALE: Heath H9 video terminal, excellent condition, $175 or best offer. You ship. **[214] 962-4484**

WANTED: DIGITAL Group 32K memory board without memory chips and Phi deck controller board (kit, assembled or not working). W... 1510 NW 35th. T... 3

PET COMPUTERS moving up to LSI-11. Pet business system priced to sell. PET 2001-16N Computer $800; 2040 Dual Floppy 340K (holds more data than 6 TRS-80 disks) $1,100. Digital cassettes (2) $60 each. System complete with Text Editor, disk sort, database software, real estate software and more $2,100. Call **PAUL** (313)971-8447

COMPUTER SHOPPER, the new buy, sell, and trade publication, is ready to help you with the latest information on personal, small business and large-system computers, accessories and software.

Each ad-packed issue is full of bargains you are looking for. Included are ads from individuals throughout the United States who are selling their good, pre-owned equipment just so they can trade-up to new equipment coming on the market.

But, COMPUTER SHOPPER'S bargains won't be yours unless you subscribe. This useful, money-saving publication can become your way to communicate with other buyers, sellers, and traders all over the nation.

Whether you are a hobbyist or a part-time user, COMPUTER SHOPPER will put you in touch with the nationwide computer marketplace in time for you to take advantage of bargain opportunities.

Have something to sell? A COMPUTER SHOPPER subscriber probably wants to buy it.

Looking for a part, component or even a complete system? A COMPUTER SHOPPER subscriber probably wants to sell it.

COMPUTER SHOPPER is THE marketplace for anything in computers and is read by thousands of people who are ready to buy.

COMPUTER SHOPPER offers a unique format in which classified ads are categorized for fast location of specific items. Combining this with low individual ad rates — 10 cents a word — makes it the ideal place for buyers and sellers to communicate. And, its mix of individual, dealer, and manufacturer ads enable subscribers to find what they want at the best price possible.

COMPUTER SHOPPER will work for you in other ways, too. If you are just thinking about getting into computers, it can help you learn product availability and prices before you make a decision. And, through the timely ads, COMPUTER SHOPPER will keep you abreast of changes in the market which could create bargain opportunities for you.

BUT COMPUTER SHOPPER cannot work for you unless you subscribe.

Want to look us over first? We'll give you your first issue FREE and then bill you for the next 12. If you are not convinced COMPUTER SHOPPER suits your needs, just write "cancel" on the invoice and return it.

And, to let COMPUTER SHOPPER start working for you right now, with a paid subscription we'll also give you a FREE classified ad to sell your pre-owned equipment or to find equipment you want.

If you don't need to use the free classified ad now, use it anytime during your subscription.

Subscription: $10/year, 12 issues plus your first free one. Bank cards accepted. Money back guarantee.

The Nationwide Marketplace for Computer Equipment

COMPUTER SHOPPER ✓36

P.O. Box F20 • Titusville, Florida 32780
Telephone 305-269-3211

EXATRON STRINGY FLOPPY Owners Association Newsletter

Secretary, Fred Waters

WHAT'S A STRINGY FLOPPY?

This little Magic Box has been around for a while now, but personal computing continues to grow at an astounding rate, and there are a lot of newcomers that may not yet have heard of the Exatron Stringy Floppy–ESF for short.

The ESF is a mass storage subsystem for your microcomputer. It does basically what either an audio-type cassette or a disk system does–provide off-line storage of programs or data. When you want to save a program now in your system memory, or save data developed within a program, such as the names and addresses that make up a mailing list program, or the record of past bowling scores and averages for a bowling league secretary's program, you need a convenient way to do this. You may have discovered by now that cassettes are slow, sometimes difficult to load and save, cumbersome to verify, and generally less reliable and convenient than you'd like. You probably have already noticed that disk systems, although fast and more reliable, are quite costly, often to the point of requiring an outlay more than the total cost of your system to date.

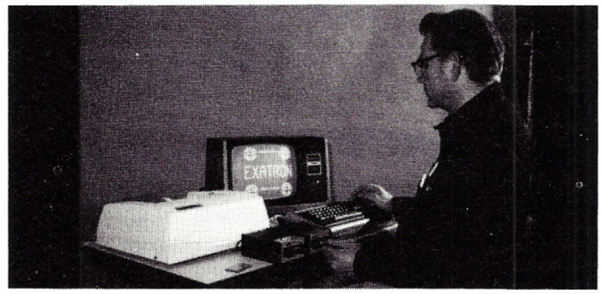
Well, the ESF gives you a great leap forward in reliability and convenience over the cassette, at something less than a quarter the cost of a disk system. There are three versions now being marketed–for S-100 systems, for the SS-50 bus, and for the TRS-80. The original version was the S-100, introduced at the 2nd West Coast Computer Faire in February 1978. Your secretary has been using this version with his IMSAI 8080–with of course successive product improvements–for about two and a half years now. Jim Maynard in Oklahoma liked the ESF, but wanted one for his SWTP micro and the 6800 CPU. So he designed it, came to Santa Clara in early 1979 to complete the development, and as Exatron project manager for the SS-50 version, introduced it at the 4th West Coast Computer Faire in May 1979. Meanwhile the demand for a TRS-80 version swelled; this version also was developed in the spring of '79, and was likewise introduced at the Faire in May. All three versions are currently available, with a 6502 version in the mill.

HOW DOES IT WORK?

The ESF consists of small case, about 4"x6"x2½", with a drive slot in the front face, two LED indicator lights, and inside the drive mechanism, tape read and write heads, and some electronics. The S-100 and SS-50 versions have ribbon cable connections to a controller board on the motherboard, and are thereby powered and controlled. The TRS-80 version has its own wall-plug transformer and power and all the firmware and control electronics are in the case, with the drive. The storage medium is digital-quality magnetic tape on a continuous loop within a miniature cartridge called a wafer. The wafer is about 3/16" thick and smaller than a business card. Tape lengths vary from 5 to 75 feet. Single-density storage and loading handles 4K bytes on 5 feet of tape in 6 seconds. That's 7200 baud, as compared to 500 for the standard cassette machine. This means that the program you now load in about two minutes–with three or four minutes more for rewind and verification–can be accurately and reliably loaded in about nine seconds! Fred Blechman, Canoga Park, CA, author and inventor and one of our most articulate boosters, says, "The simplicity and speed of operation make the Stringy Floppy compare to a cassette recorder like a modern car compares to a horse and buggy!

All ESF operations are controlled by software commands–there are no switches, buttons or other physical controls. Two modes of error detection are designed into the controls for loading. Byte-by-byte verification is provided after any save operation. New tapes are verified from end to end to ensure error-free operation. Multiple programs can be loaded onto one wafer, and called up individually by file number. To put it simply, the ESF is easy to use, most convenient, and highly reliable. IT REALLY WORKS GREAT!

But look–we're getting carried away. We can only begin to tell you here about all the features–double-density for instance in both the S-100 and SS-50 versions, and in the works for the TRS-80 version–so call us at the toll-free number below and ask for our ample information packet.



Pictured above is Bill Burnham, from Redwood City, California; just a few miles north of EXATRON here in Santa Clara. Bill is an electronics technician specializing in electronic musical instruments.

Bill says it was the Stringy Floppy that really created his inspirations to write, as it made programming so easy and fast that it now is all a labor of love.

THE WORKSHOP PROGRAM

Since January 1978 there has been a Saturday morning workshop at the Exatron plant for anyone owning or interested in the ESF. This idea has been so remarkably successful that it had to spread. Meanwhile the technical and commercial track record of the ESF has resulted in ownership of at least several ESFs in every significant population center in the U.S. Owners were asked about the workshop idea, and we now have a network of Workshop Program Chairmen all over the country. If you call our toll-free number below, not only will you get the info packet, you will also get the name of the nearest Workshop Chairman, who is prepared to answer your questions about the ESF, and with notice to conduct an informal workshop meeting and demonstrate the Stringy Floppy in action.

MICRO-COM ACQUISITION

Until recently Exatron production and expansion has been limited or blocked by dependency on the vendor of the drive and the wafers – they just couldn't keep up with the ultimate customer demand. You owners are building up libraries of software on ESF wafers, and you need more of them. You newcomers to personal computing are pushing production to satisfy the demand for the ESF. Whadda y'do in a spot like this? Acquire 'em! In February Exatron closed the deal to acquire the capital assets of Micro Communications Corp. in Waltham, MA; as of the end of February the wafer backlog is getting under control and ESF production and delivery is being accelerated. By the time you read this our hotline response to your question "How long before I can get it?" will be a reasonable one.

INFORMATION & ORDERS

The ESF is assembled and tested at the factory, with a 30-day moneyback guarantee and a one year full warranty. For fastest delivery, phone in your credit card or COD order using the toll-free line below.

Base price for the TRS-80 ESF, $249.50 (ask about the Starter Kit); for the S-100 ESF, $289.50; and for the complete SS-50 package (described in detail in last month's N/L), $499. Info packets at no charge; users manuals for the TRS-80 ESF are available for $3.00 shipping.

Handling is extra.

If you have any questions about these products, about Exatron or about ESFOA call the Hot Line. Address letters to ESFOA, 3555 Ryder St., Santa Clara, CA 95051.

HOT LINE **800-538-8559**

WITHIN CALIFORNIA **408-737-7111**

EXATRON STRINGY FLOPPY
WORKSHOP CHAIRMEN

Listed below are the names and phone numbers of ESFOA Workshop Program Chairmen. If you have any questions about the Stringy Floppy or wish a demonstration of the equipment, please contact the chairman for your area.

WORKSHOP CHAIRMEN (Listed by Zip Code)

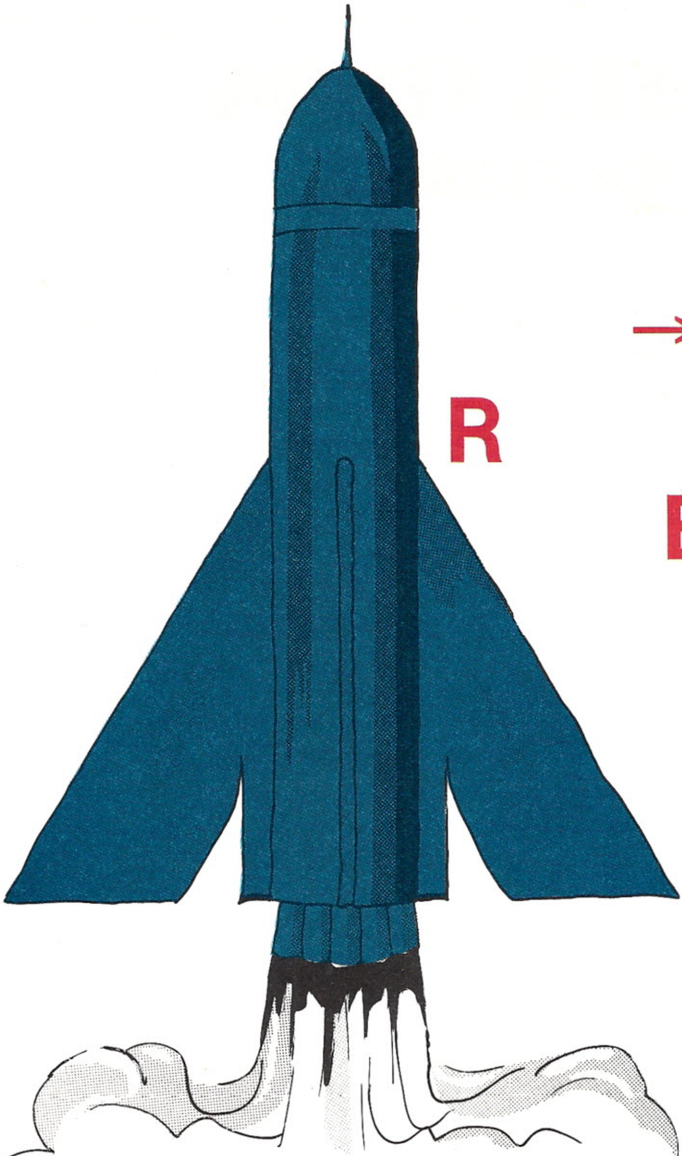
Jack O'Connell, 7 Royal Crest Drive, Marlboro, MA 01752 (617) 481-2417
Austin McCollough, 24 Donna Rd., Chelmsford, MA 01824 (617) 256-0473
R.M. Knowles, M.D, 88 Foreside Rd., Cumberland Foreside, ME 04110 (207) 773-7261
Wesley R. Day, 86 Merryfield Ave., Waterville, ME 04901 (207) 873-3955
Rev. Richard R. Noyes, 118 Main Street, Madison, ME 04950 (207) 696-5929
Michael Keller, RFD 1, Box 507, Solon, ME 04979 (207) 643-2437
Thomas Penisch, 310 Palisade Ave., Bogota, NJ 07603 (201) 343-3828
Ted Kingston, Willard Rd., Strathmere, NJ 08248 (609) 263-2440
Milton F. Thrasher, 80 Fairway Ave., Rye, NY 10580 (914) 967-3322
John E. Sheil, 196 Blacksmith Rd., Levittown, NY 11756 (516) 731-6467
HYNX Company, P.O. Box 577, Newtonville, NY 12128 (518) 783-9495
Bill Firm, 244 Albany Ave., Kingston, NY 12401 (914) 383-1824
George C. Hadley, Box 10, Paul Smiths, NY 12970 (518) 327-6413
Richard C. McGarvey, 221 Hirschfield Dr., Williamsville, NY 14221 (716) 634-3026
Robert C. Wynne, P.O. Box 704 F.S., Niagra Falls, NY 14303 (716) 285-9391
Gary E. Snyder, 37 Sumner Park, Rochester, NY 14607 (716) 461-1364
David L. MacAdam, 68 Hammond St., Rochester, NY 14615 (716) 865-8917
Paul Sturpe, 220 Wessex Hills Dr., Coraopolis, PA 15108 (412) 264-1492
Bill Dalesandry, 355 Rockfield Rd., Pittsburgh, PA 15243 (412) 221-1968
Carl A. Kollar, 1202 Gemini St., Nonticoke, PA 18634 (717) 675-5377
Bruce A. McIntyre, Rd 1, Bridgetown Pike, Langhorne, PA 19047 (215) 322-1895
Richard S. Landy, 1008 Evergreen Rd., Morrisville, PA 19067 (215) 736-0564
Robert G. Kelch, 23 Mapledale Ave., Glen Burnie, MD 21061 (301) 760-4793
Charles A. Sutton, 213 Marshall St., Salisbury, MD 21801 (301) 742-9318
Richard Harrison, Rt. 2 Drysdale, Warrenton, VA 22186 (703) 439-8109
William P. Schultze, 931 Branchwater St., Fredericksburg, VA 22401 (703) 786-8878
William C. Geer, Rte 5, Box 502, Falmouth, VA 22401 (703) 371-2019
H. Ross Wiant, 240 Brook La., Grafton, VA 23692 (804) 898-8120
Archie H. Hubbard III, 520 Lawrence Ave., Bristol, VA 24201 (703) 669-3191
Lawrence P. Doyle, 717 Patrick St., Eden, NC 27288 (919) 627-0359
Micalel W. Talbot, 3601 Carriage Dr., Raleigh, NC 27612 (919) 787-8082
Herbert Alperin, 5675 Peachtree Dunwoody Rd NE, Atlanta, GA 30342 (404) 256-3414
Mark C. Few, 2827 Parkland Cir., Orlando, FL 32803 (305) 644-7464
John T. Charlton, 688 Caribbean Rd., Sat Beach, FL 32937 (305) 773-1590
Richard B. Nelson, 763 W. 63rd St. Ocean, Marathon, FL 33050 (305) 743-9195
Frank Lungarella, 648 DeLeon Dr., Miami Springs, FL 33166 (305) 887-4821
A.J. Catano, 11810 NW 29th Pl., Sunrise, FL 33323 (305) 741-4325
Robert W. Hopkins, 212 31st St., West, Bradenton, FL 33505 (813) 748-7074
William C. Howes, 5597 Trellis Ln., Fort Myers, FL 33907 (813) 482-0347
Bob Jones, 4084 Berrybrook Cove, Memphis, TN 38138 (901) 363-5469
Sumner Loomis, Rt. 1, Box 131, Prairie Point, MS 39353 (601) 726-5524
Lawrence W. Stein, 867 Independence Station Rd., Independence, KY 41051 (606) 371-3242
W.H. Jenkins, 33 W. Columbus St., Canal Winchester, OH 43110 (614) 837-2325
Lester R. Snowden, 1821 Pine St., Rafb, OH 43217 (614) 497-0838
Paul Schmid, 201 Mohican Ave., Orrville, OH 44667 (216) 682-3596
Robert E. Morgan, 1127 Princeton, Celina, OH 45822 (419) 586-7416
Don Heringhaus, 1055 Woodside Dr., Ottawa, OH 45875 (419) 523-5613
William R. Gouge, 6428 Topeka Ln, Indianapolis, IN 46241 (317) 856-6339
Jack J. Fleming, 322 Jeffrey, Royal Oak, MI 48073 (313) 435-4487
Robert M. Clark, 20515 Gaylord, Redford, MI 48240 (313) 535-5722
Dennis Chaney, 1239 S. Ruess, Owosso, MI 48867 (517) 723-1011
Alan C. Lanphear, 727 Ray St., Kalamazoo, MI 49005 (313) 344-8277
Ranjit Diol, RR6, Kilburn Ave., Dowagiac, MI 49047 (616) 424-5440
Donald Meinke, 2507 Blaine SE, Grand Rapids, MI 49507 (616) 245-9420
R.L. Knapp, Jr., 414 2nd St., Perry, IA 50220 (515) 465-5873
Robert F. Gerald, 1377 Arden View Dr., Arden Hills, MN 55112 (612) 636-0549
Robert W. Foster, 791 White Birch Ln, Lake Zurich, IL 60047 (312) 438-5223
Martin Berger, 4634 Oakton St., Skokie, IL 60076 (312) 674-3038
Joel Natkin, 1626 Keeney, Evanston, IL 60202 (312) 869-1660
Gary P. Cocanour, 333 Central Ave., Hinsdale, IL 60521 (312) 986-1059
James E. Spath, 3610 N. Pinegrove Apt. 301, Chicago, IL 60613 (312) 353-2544
Richard Kramer, 401 N. Cherry, Abingdon, IL 61410 (309) 462-2859
John M. Delaney Jr., 860 Penning Ave., Wood River, IL 62095 (618) 254-4571
Donald L. Hiles, 802 Grand Ave., Beardstown, IL 62618 (217) 323-3963
Lynn A. Nelson, 409 San Jose, O'fallon, MO 63366 (314) 272-1153
James G. Runyan, 1400 N. Pine Apt. 6, Rolla, MO 65401 (314) 364-7306
Ed Forest, 8224 Tomahawk Rd., Prairie Village, KS 66208 (913) 648-1513
Charles L. Perrin, 5001 E. Pawnee 113B, Wichita, KS 67218 (316) 685-0528
Val Lindholm, Box 1105, 5512 Comanche, Great Bend, KS 67530 (316) 793-9319
Robert K. Lahm, 224 4th Ave., Holdrege, NE 68949 (308) 995-4071
R.M. Wright, 3416 Greenbriar Ln, Plano, TX 75074 (214) 596-7000
Dr. Murlon H. Dye, 1412 Locust St., Commerce, TX 75428 (214) 886-6434
Donald Michel, 2601 Rushvalley Ct., Arlington, TX 76016 (817) 469-1521
W.L. Lonnon, Rt 3, Box 49-1, Decatur, TX 76234 (817) 627-3838
Kent Zakrzewski, 619 Cortlandt, Houston, TX 77007 (713) 448-6374
George Lyda, Box 127, New Baden, TX 77870 (713) 828-4523
Stephen E. Fisch, 5225 Moultrie, Corpus Christi, TX 78413 (512) 991-4584
R. B. Sexton, 2200 W. Anderson Ln., Austin, TX 78757 (512) 452-6131
John Dierdorf, 8109 Greenslope Dr., Austin, TX 78759 (512) 345-3912
Richard Laubhan, 111 N. Claremont, Colorado Springs, CO 80909 (303) 632-7580
Dennis E. Murray, 989 E. 1050N. Ogden, UT 84404 (801) 782-8042
John Crutcher, 6015 E. Pershing Ave., Scottsdale, AZ 85254 (602) 948-2911
Bob Rankin, 1018 W. Hermoja Dr., Tempe, AZ 85282 (602) 967-7677
Robert D. Sullivan, 1537 W. 7th St. No. 120, Upland, CA 91786 (714) 985-5273
Frank M. Harris, 4570 Ohio St., San Diego, CA 92116 (714) 281-9286
Donald Haskell, 4404 W. Hurley Ave., Visalia, CA 93277 (209) 732-1553
Hubert C. Wood, 201 McCord, Sp. 5, Bakersfield, CA 93308 (805) 399-8252
Bill Burnham, 3659 Glenwood Ave., Redwood City, CA 94062 (415) 365-0760
G. K. Johnson, 1009 Aspen Way, Petaluma, CA 94952 (707) 762-2168
Dustin F. Leer, Box 1664, Sausalito, CA 94965 (415) 332-2557
James Ferguson, 2485 Autumnvale Ave., San Jose, CA 95132 (408) 946-1265
Lars A. Benson, 436 Klute St., No. 2, Santa Rosa, CA 95401 (707) 523-1782
Richard D. Mack, 1748 Fenwick Dr., Santa Rosa, CA 95401 (707) 528-0309
J. J. Waters, 333 LaCrosse Ave., Santa Rosa, CA 95405 (707) 539-2595
Vaughn A. Jupe, RFD Box 60, Carlotta, CA 95528 (707) 768-3833
E.R. Hubbard, 2558 Deertrail Ln., Shingle Springs, CA 95682 (916) 677-5440
Paul F. LeFevre, 3529 Cody Wy 109, Sacramento, CA 95825 (916) 483-4760
Alfred J. Kiep, Jr., 605 Wilcox, Ave., Etna, CA 96027 (916) 467-3255
Gilbert Olayan, 961 Apokula St., Kailua, HI 96734 (808) 261-2228
L.P. Folz, 419 Madrona SE, Salem, OR 97302
Ken Ernst, 3921 Ibex NE, Salem, OR 97303 (503) 393-1173
J.R. Shunn, 2625 Altamont No. 10, Klamath Falls, OR 97601 (503) 882-2023
Rick Winter, P.O. Box 933, Pendleton, OR 97801 (503) 276-3611
J.A. Records, 1106 Cedar St., LaGrande, OR 97850 (503) 963-0437
Ralph Baker, P.O. Box 250, Long Beach, WA 98631 (503) 642-3126
Allan Stubbs, 1598 Cedar St., Prince George, B.C. Canada, V2L-1B7 563-6894

6800 SS-50 ESF OWNERS

John Drum, 44 County Village Ln., Sudbury, ME 01766
John Pyra, 44 Flagg Rd., Westford, ME 01866
James Robertson, 21 Grafton Dr., Morris Plains, NJ 07950
G.S. Lyons, 217 Homestead Ave., Huddonfield, NJ 08033
Bits Computer Systems, 337 Geln Rd., Rome, NY 13440
Richard Kerns, 616 Beatty Rd., Monroe, PA 15146
Mike Mellon, 211 San Paulo Cir, W. Melbourne, FL 32901
Norman Weiner, 1700 Convention Center, Miami Beach, FL 33139
Gordon Doughman, 4250 Moselle Dr., Hamilton, OH 45011
Paul Manter, 10665 Stargate Ln., Cincinnati, OH 45240
Garrett Smith, 2001 NW Hickory Ln, No. 9, Ankery, IA 50021
Mark Begemann, 1985 24th St., Mario, IA 52302
Ray Westberg, 5702 34th Ave., N. Minneapolis, MN 52422
Rodney Schnueland, 9027 Pierce St. NE, Blaine, MN 55434
Laurence Drymalski, 1232 Countryside Dr., Elgin, IL 60120
Donald Miser, 1015W Wood, Bloomington, IL 61701
John Ramsey, 8307 E. 166th St., Belton, MO 64012
Al Scherer, M.D., 610 E. 1st St., St. John, KS 67576
Wade Maxfield, 2525 E. Union Bower, Evering, TX 75061
Richard Poindexter, 4347 Phelan St., Beaumont, TX 77707
John Stimm, 6220 Calebra Rd, San Antonio, TX 78284
European Auto Parts, 4322 B.N. Lamars, Austin, TX 78756
John Lutenberg, 923 Main St., Canyon City, CO 81212
UC Los Alamos Scientific Lab, Los Alamos, NM 87545
Thomas Mattingly, 1005 Essex Dr. W. , Las Vegas, NV 89107
Robin Ellwood, 18506 Dylan St., Northridge, CA 91326
Don McPhee, 1550 4th Ave., S., Seattle, WA 98134
P. Peterson, 9117 NE 96th St., Vancover, WA 98662
Jerry Starzinski, 58th Ave., Yakima, WA 98903
Motorola, 2400 S. Rossevelt, Tempe, AR
J.L. Bauliane, 956 Lillian St., Coquittam, British Columbia, CAN V3J 5C4

ORBITER

TRS-80 Level II



Terry Kepner
Bud Grace
P O Box 481
Peterborough NH 03458

Ever since I first used an interactive computer, I have been playing different versions of the Lunar Lander game (lunar excursion module). All of these versions started from the same premise: The LEM is x miles/feet away from the landing sight and y miles/feet above the surface, with a horizontal velocity of dx feet per second.

None of these versions has really approached a true simulation. Even when I was using the DEC-10, the best version of the game began in this manner, although it did restrict the player to only those parameters and thrust values used aboard the actual LEM.

After having an R/S L II for almost a year, I decided to try to program a LEM simulation that not only retained the true parameters, but also started out in a true orbit from which the player had to try to land the LEM.

Two weeks later my room was waist deep in paper, and I had run out of hair to pull out in frustration. I just couldn't make my equations work properly, and none of my textbooks helped. At about this time I mentioned my difficulties to a friend of mine, Bud Grace. Eureka!

Not only could he help me with the equations, but he already had a program capable of evaluating orbits. The program, cleaned up and with REM statements added, is shown in Listing 1.

How the Programs Work

Upon inspection you'll note that the gravity of the central sun (or black hole, if you prefer) is set at 50 times that of the Earth, and the acceleration of the rocket's engines is set at one half that of the Earth's gravity. For simplicity and to speed up the computations, the mass of the central sun and the rocket are assumed to be equal, unit mass one. Also, instead of using a differential equation, a simple summation process is used,

```
5 REM ORBITER PROGRAM
10 CLS
15 REM TURN ON SUN IN SCREEN CENTER
20 SET(63,23)
25 REM SET ENGINES (IN EARTH GRAVITIES)
30 A=.5
35 REM SET SUN'S GRAVITY
40 G=50
45 REM SET INITIAL SHIP POSITION
50 X=-30:Y=-20
55 REM CHECK KEYBOARD FOR INPUT
60 K$=INKEY$
65 REM COMPUTE CHANGES INTRODUCED BY ENGINES
70 IF K$="J" THEN DX=DX+A
80 IF K$="F" THEN DX=DX-A
90 IF K$="B" THEN DY=DY+A
100 IF K$="Y" THEN DY=DY-A
105 REM COMPUTE SUN'S PULL ON SHIP
110 X1=X*X
120 Y1=Y*Y
130 Z1=X1+Y1
140 D=G/(Z1)[(3/2)
150 FX=-X*D
160 FY=-Y*D
165 REM COMPUTE CHANGES IN SHIP'S COURSE
170 DX=DX+FX
180 DY=DY+FY
185 REM TURN OFF OLD POSITION ON SCREEN
210 RESET(H,V)
215 REM COMPUTE NEW POSITION
220 X=X+DX
230 Y=Y+DY
235 REM TURN ON NEW SCREEN POSITION
240 H=2*X+63
250 V=Y+23
260 SET(H,V)
270 GOTO 60
```

Listing 1. Orbiter program.

again for speed in running. Despite these shortcuts the orbits established can be stable for over a 1000 revolutions.

The game starts with the rocket in the upper left-hand corner of the screen, at a dead stop with relation to the central sun. It will free-fall toward the sun, as you hit the proper keys to fire the engines and try to set up an orbit.

One other point to notice is that the position of the rocket on the screen has been corrected for the difference in x-y axis spacing (lines 240 and 250); that is, a circle on the screen is really a circle and not an ellipse distorted to appear as a circle. And last, the program crashes if the rocket tries to leave the screen.

Listing 2 is an improved version of the orbiter changed into a simple game. The screen limitation has been removed, and the x-y coordinates of the rocket are displayed in the upper left corner of the screen whenever the rocket does leave the screen. For ease of use, the direction keys have been changed to use the keyboard arrows.

The game feature involves the use of orbital bombs created by the computer and dropped onto the screen with random velocities. If one of these bombs comes within range of the rocket, the rocket will blow it up. Only one bomb is allowed on the screen at a time. To control the frequency that these bombs appear, a number between 1 and 50 is chosen by the player at the beginning of the game. Choosing 1 ensures that there will always be one bomb on the screen, while choosing the 50 means that once every 50 screen updates a bomb should appear, on the average.

Referring back to Listing 1, you might try removing the reset (line 210) and watching how the orbits precess around the sun. When this is done, you will notice that no matter how elliptical the orbit may appear at first, as time passes, the orbital pattern is symmetrical. This precession is normal for all orbits and may require as many as 2000 orbits to precess in a complete 360° circle. ■

```
10 CLS
20 REM BY BUD GRACE
50 PRINT"ORBITER"
60 PRINT
65 REM  DIRECTION CONTROLS
70 PRINTCHR$(91);" = UP"
80 PRINTCHR$(92);" = DOWN"
90 PRINTCHR$(93);" = LEFT"
100 PRINTCHR$(94);" = RIGHT"
110 HO=40
120 VE=30
140 PRINT
145 REM THIS CONTROLS THE BOMBS  1=ALWAYS A BOMB, 50=RARELY
150 INPUT"WHAT LEVEL OF DIFFICULTY DO YOU WANT (1 - 50)";TY
155 REM SET CONSTANTS AND INITIALIZE THE VARIABLES
160 Q=3/2
170 E=31
180 B=23
190 C=63
200 R=2
210 CLS
215 REM SET THE SUN
220 SET(C,B)
230 A=.5
240 G=30
250 X=-29
260 Y=-20
265 REM CHECK KEYBOARD FOR INPUT
270 K$=INKEY$:IFK$=""THEN320ELSEK=ASC(K$)
275 REM COMPUTE CHANGES INTRODUCED BY KEYBOARD
280 IFK=9THENDX=DX+A
290 IFK=8THENDX=DX-A
300 IFK=10THENDY=DY+A
310 IFK=91THENDY=DY-A
320 SW=X
330 SU=Y
335 REM CALCULATE ACCELERATION DUE TO GRAVITY
340 GOSUB740
345 REM UPDATE VARIABLES
350 FX=-X*D
360 FY=-Y*D
370 DX=DX+FX
380 DY=DY+FY
385 REM IF THERE ISN'T A BOMB, THEN IS IT TIME FOR ONE?
390 IFFLAG=0THENTEST=RND(50*TY/50)
400 IFFLAG=1ORTEST<>1THEN460
405 REM IF IT IS TIME, THEN CREATE ONE
410 HO=RND(C)-E
420 VE=-B
430 DH=RND(3)-R
440 DV=RND(5)-3
450 FLAG=1
455 REM IS THERE A BOMB ON THE SCREEN
460 IFFLAG=0GOTO660
465 REM IF SO, THEN UPDATE ITS POSITION
470 SW=HO
480 SU=VE
485 REM ACCELERATE THE BOMB
490 GOSUB740
500 DH=DH+HO*D
510 DV=DV+VE*D
520 HO=HO-DH
530 VE=VE-DV
535 REM IF ITS NEW POSITION IS OFF SCREEN, DESTROY IT
540 IFABS(HO)>EORABS(VE)>BTHENFLAG=0:RESET(H1,V1):GOTO660
545 REM TURN OFF BOMB'S OLD MARKER
550 RESET(H1,V1)
560 H1=R*HO+C
570 V1=VE+B
575 REM TURN ON BOMB'S NEW MARKER
580 SET(H1,V1)
585 REM TEST TO SEE IF ASTRONAUT IS WITHIN RANGE
590 IF(ABS(HO-X))>4OR(ABS(VE-Y))>2THENGOTO660
595 REM IF YES THEN BLOW HIM UP
600 PRINT@0,"YOU ARE A  DEAD ASTRONAUT -- YOU SHOULD BE MORE CAREFUL"
610 PRINT"DO YOU WANT TO TRY AGAIN?";
615 GOSUB760
620 K$=INKEY$
630 IFK$=""THEN620
640 IFK$="Y"ORK$="YES"THENRUN
650 END
655 REM IF THE SHIP IS ON THE SCREEN, TURN OFF ITS MARKER
660 IFABS(X)<EANDABS(Y)<BTHENRESET(H,V)ELSEPRINT@0,CHR$(30)
670 X=X+DX
680 Y=Y+DY
690 H=R*X+C
700 V=Y+B
705 REM TURN ON ITS NEW MARKER
710 IFABS(X)<EANDABS(Y)<BTHENSET(H,V)ELSEPRINT@0,INT(X),INT(Y)
715 REM IF TOO CLOSE TO SUN, DESTROY SHIP
720 IFABS(X-C)<4ANDABS(Y-B)<2THEN600
730 GOTO270
735 REM THIS COMPUTES THE EFFECT OF GRAVITY
740 D=(G/(SW*SW+SU*SU))[Q
750 RETURN
755 REM THIS EXPLODES THE SHIP
760 FORJ=H-3TOH+3STEP2
770 FORI=V-2TOV+2STEP2
780 SET(J,I)
790 NEXTI,J
800 RETURN
```

Listing 2. Improved version.

The Personal Computer Line by OHIO SCIENTIFIC

Personal Computers

C1P: $349 A dramatic breakthrough in price and performance. Features OSI's ultra-fast BASIC-in-ROM, full graphics display capability, and large library of software on cassette and disk, including entertainment programs, personal finance, small business, and home applications. It's a complete programmable computer system ready to go. Just plug-in a video monitor or TV through an RF converter, and be up and running. 15K total memory including 8K BASIC and 4K RAM—expandable to 8K.

C1P MF: $995 First floppy disk based computer for under $1000! Same great features as the C1P plus more memory and instant program and data retrieval. Can be expanded to 32K static RAM and a second mini-floppy. It also supports a printer, modem, real time clock, and AC remote interface, as well as OS-65D V3.0 development disk operating system.

Professional Portables

C4P: $698 The professional portable that has over three times the display capability of C1Ps. Features 32 x 64 character display in up to 16 *colors*, graphics, audio output, a DAC for voice and music generation, key pad and joystick interfaces, AC remote control interface and much more. Utilizes a 4-slot BUS (2 used in base machine), 8K BASIC-in-ROM, 8K of *static* RAM and audio cassette interface. Can be directly expanded to 32K static RAM and two mini-floppy disks.

C4P MF: $1695 The ultimate portable computer has all the features of the C4P plus real time clock, home security system interface, modem interface, printer interface, 16 parallel lines and an accessory BUS. The standard machine operates at twice the speed of currently available personal computers (with GT option it runs even faster!). The C4P MF starts with 24K RAM and a single mini-floppy and can be directly expanded to 48K and two mini-floppies. Available software includes games, personal, business, educational and home control applications programs as well as a real time operating system, word processor and a data base management system.

Computers come with keyboards and floppies where specified. Other equipment shown is optional.

Home/Small Business Systems

C8P: $895 Same great features as the C4P in a tremendously expandable "mainframe package." Features over three times the expansion capability of the C4P for advanced home and demanding business applications. Can be expanded to 48K RAM, dual 8" floppies, hard (Winchester) disks and multiple I/O devices such as Voice I/O and a universal telephone interface.

C8P DF: From $2597 The ultimate Home/Very Small Business Computer at a personal computer price. Features 32K RAM (expandable to 48K) and dual 8" floppy disks (stores eight times as much information as a mini-floppy). Has all personal computer capabilities including 32 x 64 display, color graphics, sound, DAC, joystick interfaces, home features including real time clock, AC remote interface, home security and fire detection interface and can be expanded to include voice I/O and a universal telephone system for answering and initiating calls! Its large memory capability and 8" floppies allow it to run most Ohio Scientific business system software including a compete accounting system, word processor and information management system.

For literature and the name of your local dealer, CALL 1-800-321-6850 TOLL FREE.

OHIO SCIENTIFIC
1333 SOUTH CHILLICOTHE ROAD
AURORA, OH 44202 • (216) 831-5600

5

Wire Listings the Easy Way

Yes, even a constructional klutz can wire-wrap successfully.

Jack G. Sheppard, Ph.D., P.E.
5242 Laguna
Sierra Vista AZ 85635

I always solder connectors on both ends of the cable before slipping the shells on. Last night I even discovered a variant of the procedure. Forced by the Lady of the Manor to move the computer from the living room, I had to install a stereo outlet in the wall of its new quarters.

Immersed with total concentration in the project, I mated a phone jack to the new cable and looked around for the wall plate. It was on the wall. The wires on which I had just soldered the phone jack protruded neatly through its hole. The phone jack, which is supposed to mount behind the plate, is larger than the hole in the plate.

Did I unsolder the phone jack and calmly reinstall it correctly, or did I break the plate? I can testify that the plastic wall plates shatter most satisfyingly when attacked with diagonal cutters.

Introduction

There are lots of us computer owners who are afflicted with constructional klutzomania. At the same time, there are lots of gadgets we would like to have for our computers that we probably can't afford to buy. Our only alternative is to build them.

And why not? Microcircuits virtually paper the walls of every radio and computer store. If you hook a few together, they will do anything you want. Designing the logical circuit is no problem . . . certainly no more difficult than designing the programs we feed to our computers.

But think of all those wires! Even a small project will necessitate several hundred connections. And what about the klutzomaniac?

The rest of this article describes a program that generates easy-to-follow wire-connection lists. I have completed two major construction projects using this program without a single wiring error.

The Program

The program was written for the Level II TRS-80, although it can be adapted to any computer that allows arrays. It will prompt you through the process of developing the listing so that you have a minimum chance of making errors. It keeps track of what chip you are working on and which pins have already been treated, so that you don't end up wiring the same circuit twice. It will also ensure that no pin is ignored. At the end, you will have a list of pin-pairs on the construction board that you can wire directly with confidence.

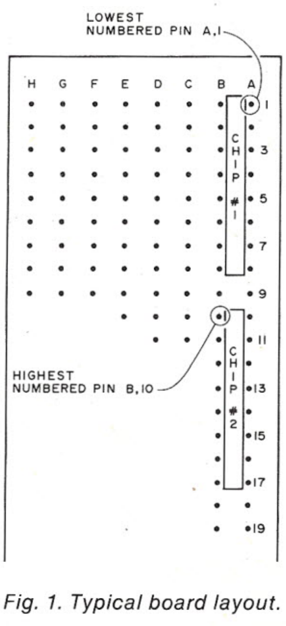


Fig. 1. Typical board layout.

Begin any construction project with a clean logic diagram on which each chip, along with each of its pins, is displayed. Number each chip. Show the wiring destination of each pin, either by drawing in the wire or by listing destination chips and pins.

Next, lay out the construction board. Chips should be located (not installed at this point) on the board in positions that will minimize wiring runs. Label columns of pins using letters, and label rows of pins using numbers (this is the format of an Augat wire-wrap board, which is the construction medium that I prefer).

Columns of pins on the board should align with the pins of the chips to be installed. Thus, pins 1–8 of chip 1, for example, may occupy board positions A1 through A8 (see Fig. 1). Identify and record the positions on the board of pin 1 and the highest-numbered pin of each chip. You are now ready to run the program and develop your wire list.

Operation

After you load the program, it will ask how many chips you will be using and what the largest pin-count per chip is. This information is used to set up arrays on which the program will operate. The program will then initiate a loop that asks for the number of pins on each chip and the board locations of the lowest- and the highest-numbered pins. This information is used to set the locations of the chips with respect to the pin matrix used on the construction board. This loop also converts the column-designating letter into an equivalent number for use by the computer.

I wanted the program to guarantee that I didn't overlook any pin or wire any pin twice unless it needed it. I set up a two-dimensional array, E(I,J), to keep track of the pins used at any point in program execution. A pair of nested loops ensures that all values are zero at the outset. The program marks each pin with a nonzero value in this array as it is used.

Next, a pair of nested loops begins with chip 1, pin 1, and performs the following functions:

1. Checks whether the pin has been used. If so, goes to the next pin.

2. Asks for connection data with the statement "CHIP 1 PIN 1 GOES TO (CHIP,PIN)?" If there is no connection, enter "0,0", and the loop marks the pin and proceeds to the next pin. If the pin is to be connected, you should respond with the correct destination chip and pin. The program will mark the destination pin as having been used and print the linkage. It will then ask for any additional connections to chip 1, pin 1. You should enter all paths that can be traced to this pin. After all connections have been logged, enter "0,0". The program will mark chip 1, pin 1, "used" and

advance to chip 1, pin 2.

3. Data is entered as chip numbers and pin numbers. The construction board is laid out in lettered columns and numbered rows. Chip 7, pin 13, may be located at column F, row 45, for example. The loop converts the chip #, pin #, into construction-board format. There is one tricky operation, highly prone to manual error, and is the reason I developed this program in the first place. On a 16-pin chip, pins 9-16 number in the reverse direction from the row numbers on the construction board. In converting to board format, you must determine whether the pin is in the upper half or lower half, so that you know which column to use and whether to add or subtract to get the row number. My printer does not have less-than/greater-than symbols, so I used brackets ([,]) instead.

4. As each data-pair is logged and converted into column and row form, it is printed on the CRT. The results are not stored in retrievable form, so you will need to copy them from the CRT as they are displayed. If you have a printer, modify statement 290 to LPRINT, and you will not need to copy the data manually.

5. This loop is not an automatic loop. It sticks with one pin until the user enters "0,0". Then it goes on to the next pin, and so on until all pins have been used and marked.

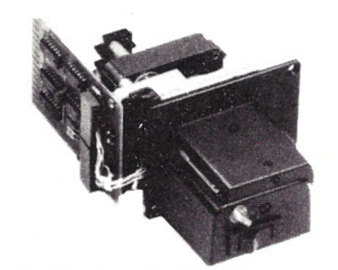
Conclusion

A large construction project requires that you key in a large number of data values. A computer cannot read minds and cannot identify errors that fit the prescribed form, so check the data as you key it in. If you do that and follow the resultant list scrupulously, you should have virtually error-free wiring. As a bonus, you can use a multimeter to run a quick continuity check on the wrapped board to determine whether you have put wires on the wrong pins, or vice versa. ■

```
1 REM   --THIS PROGRAM CREATES WIRE LISTS--
10 INPUT "WHAT IS THE LARGEST PIN COUNT?";A1
15 REM  --INPUT TOTAL NUMBER OF IC'S USED IN PROJECT--
20 INPUT "HOW MANY CHIPS?";A
30 DIM B(A),C1(A),C2(A),D1(A),D2(A),E(A,A1)
35 REM  --LOOP TO SET UP LOCATION DATA FOR ALL CHIPS--
40 FOR I=1 TO A
50 PRINT "HOW MANY PINS ON CHIP ";I;:INPUT B(I)
60 PRINT "ENTER LOCATION OF CHIP ";I; "PIN 1 (COLUMN, ROW)";
65 INPUT C$,C2(I)
70 PRINT "ENTER LOCATION OF CHIP ";I;" PIN ";B(I);
75 INPUT D$,D2(I)
76 REM  --CONVERT COLUMN-LETTER TO EQUIVALENT NUMBER--
80 C1(I)=ASC(C$)-64:D1(I)=ASC(D$)-64
90 NEXT I
95 REM  --LOOP TO ZERO ALL PIN-USE MARKERS--
100 FOR I=1 TO A:FOR J=1 TO A1:E(I,J)=0:NEXT J:NEXT I
105 REM  --CHIP SELECT LOOP--
110 FOR I=1 TO A
115 REM  --PIN SELECT LOOP--
120 FOR J=1 TO B(I)
125 REM  --CHECK WHETHER PIN HAS BEEN USED--
130 IF E(I,J)[]0 GOTO 310
135 REM  --INPUT CONNECTION DATA--
140 PRINT "CHIP ";I;" PIN ";J;" GOES TO (CHIP, PIN)";: INPUT X,Y
145 REM  --CHECK FOR CONNECTION--
150 IF X[]0 GOTO 170
155 REM  --MARK PIN USED--
160 E(I,J)=1:GOTO 310
165 REM  --MARK DESTINATION USED--
170 E(X,Y)=2
175 REM  --CHECK WHETHER SOURCE-PIN IS UPPER OR LOWER HALF--
180 IF J]B(I)/2 GOTO 210
185 REM  --LOWER-HALF CONVERSION--
190 S=C1(I)
200 U=C2(I)+J-1: GOTO 230
205 REM  --UPPER-HALF CONVERSION--
210 S=D1(I)
220 U=D2(I)+B(I)-J
225 REM  --CHECK WHETHER DESTINATION-PIN IS UPPER OR LOWER HALF--
230 IF Y]B(X)/2 GOTO 260
235 REM  --LOWER-HALF CONVERSION--
240 W=C1(X)
250 V=C2(X)+Y-1:GOTO 280
255 REM  --UPPER-HALF CONVERSION--
260 W=D1(X)
270 V=D2(X)+B(X)-Y
275 REM  --CONVERT COLUMN NUMBERS TO LETTERS--
280 S$=CHR$(S+64):W$=CHR$(W+64)
285 REM  --PRINT CONNECTION DATA--
290 PRINT S$;",";U;" CONNECTS TO ";W$;",";V
300 GOTO 130
310 NEXT J
320 NEXT I
330 END
```

Program listing.

Reader Service—see page 258



350 Char/Sec Low Cost Loader

Solid state photoelectronic components read all standard 5, 6, 7 or 8-level tapes. Smooth, quiet AC drive. Reliable high-speed paper tape data entry. Data amplifiers and "character ready" output available for CMOS or TTL interfaces. Standard interface or to your interface specifications. Fanfold boxes available.

Addmaster Corporation, 416 Junipero Serra Drive, San Gabriel, CA 91776, Telephone: (213) 285-1121, Telex: 674770 Addmaster SGAB ✓261

TRS-80
LEVEL 1 4K EDUCATIONAL SOFTWARE

ELECTRONIC SERIES:

6 programs, plus graphics, covers resistance, Ohm's law, capacitance, capacitive reactance, resonant circuits, energy and power. Cassette No. 123179KP. $9.95

METRIC SERIES:

6 programs for consumers. Each program includes computer generated tests. Covers prefixes, length, weight, volume, and temperature.
Cassette No. 122579JL. $9.95

Let us market **YOUR** programs. Either royalty or direct puchase. Send S.A.S.E. for details.

Include $1.55 for postage and handling. Send check, M.O. to.

NORKA SOFTWARE
BOX 6203
AKRON, OH 44312 ✓212

CUDDLY SOFTWARE

Unbreakable Systems/Support Software for any 1802 system.

CSOS Series (Operating System) – aids development and modification of user programs with software tools, like Add/Delete Byte, which also expand/contract the remainder of the memory page. User Programs also interface via Std. Call/Ret. for video routines, I/O drivers, etc. 1861 alphanumeric output via Screen Driver (10 lines, ave. 17 char/line, auto-scroll). User graphics also made easy.

CSTP Series (Trace Program) – 1802 CPU in software! Displays internal status as it simulates user programs for debugging, improvement, or learning 1802 operation. A must for every programmer! Intervene to alter simulation dynamically, by changing parameters, overriding program branch decisions, etc. Memory protected! CSTP cannot be bombed by the worst test program or user key-in errors! Needs 2.5K RAM, plus user program area.

COMING SOON: Assembler; Text Editor; Text Formatter; I/O for 1802 PILOT!

TAILORED to your system! Write for details and low prices.

CUDDLY SOFTWARE
Dept. K41
157 CHARTER ROAD
WETHERSFIELD, CT. 06109 ✓262

TRS-80*
SAVE A BUNDLE

When you buy your TRS-80™ equipment!

Use our toll free number to check our price before you buy a TRS-80™ . . . anywhere!

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

full Radio Shack warranty

T.A.B. SALES COMPANY ✓189
P.O. BOX 8098 PENSACOLA, FL 32505
FLORIDA 904/438-6507
nationwide 1-800-874-1551

16 K UPGRADE
$66.95
TRS80, APPLE II, AND SORCERER

HIGH QUALITY PRIME 16K RAMS FOR MEMORY UPGRADE. THE KIT INCLUDES, THE RAMS, SHUNTS AND INSTRUCTIONS TO ALLOW EASY UPGRADE IN MINUTES. ALL PARTS CARRY 12 MONTH WARRANTY.

TO ORDER, SPECIFY KIT AND ENCLOSE CHECK OR MONEY ORDER. ADD $2.00 POST AND PACKING; TEXAS RESIDENTS ADD 5% SALES TAX.

IAN ELECTRONICS
P.O. BOX 14079
AUSTIN, TEXAS 78761 ✓209

EXIDY Sorcerer 16K $1,089✦
commodore Pet 16K $799✦

Call or write for price list with comparable savings on 32/48K models and peripherals. ✓260

MicroComputer Warehouse
3620 La Habra Way
Sacramento, Ca. 95825
(916) 486-3678

- Computers are fully tested and burned in for 48 hours.
- We respond promptly to all inquiries and orders. Send $1 for catalog, refundable with order.

✦Cash price–add 2% for Visa/Mastercharge. Price subject to change without notice.

A growing line of tools to expand the Apple.

7440A Programmable Interrupt Timer Module. Time events in four operating modes—continuous, single shot, frequency comparison, and pulse width comparison. Includes three 16-bit interval timers, plus flexible patch area for external interface. Programmable interrupts, on-board ROM, and much more.

7720A Parallel Interface. Two bi-directional 8-bit I/O ports will connect your Apple to a variety of parallel devices, including printers, paper tape equipment, current relays, external on/off devices. Full featured, programmable interrupts, supports DMA daisy chaining.

7811B Arithmetic Processor. Interfaces with Applesoft, so you just plug in and run. Based on the AM 9511 device, provides full 16/32-bit arithmetic, floating point, trigonometric, logarithmic, exponential functions. Programmed I/O data transfer, much, much more.

7710A Asynchronous Serial Interface. Conforming to RS-232-C A thru E 1978 standard, this card will drive a variety of serial devices such as CRT terminals, printers, paper tape devices, or communicate with any standard RS-232 device, including other computers. Full hand-shaking, and fully compatible with Apple PASCAL!

7470A 3¾ BCD A/D Converter. Converts a DC voltage to a BCD number for computerized monitoring and analysis. Typical inputs include DC inputs from temperature or pressure transducers. Single channel A/D, 400 ms per conversion.

7490A GPIB IEEE 488 Interface. A true implementation of the IEEE 488 standard—the standard protocol for instrumentation and test devices. Control and monitor test instruments such as digital voltmeters, plotters, function generators, or any other device using the IEEE 488.

7114A PROM Module. Permits the addition to or replacement of Apple II firmware without removing the Apple II ROMs. Available with on-board enable/disable toggle switch.

7500A Wire Wrap Board. For prototyping your own designs.

7510A Solder Board.

7590A Extender Board.

7016A 16K Dynamic Memory Add-On.

Watch this space for new CCS products for the Apple. We've got some real surprises in the works. To find out more about the CCS product line, visit your local computer retailer. The CCS product line is available at over 250 locations nationally, including most that carry the Apple. Or circle the reader service number on this ad.

Apple II, Apple II Plus, and Applesoft are trademarks of the Apple Corporation.

CCS makes the difference.

We see the Apple

We see it as a good way to get things done.

Apple has built a great computer. We at CCS have built a great line of peripherals and components to expand the Apple. To do almost anything you want to get done with a computer.

If you want to do business with an Apple, we've got tools to connect the Apple to standard business printers and terminals. Or to modems, for communications over telephone lines, with other computers, even with other Apples.

If you want to apply your Apple to engineering, scientific, or graphic projects, we've got tools for high-powered, high-speed math functions, and fast, high resolution graphics. And tools to connect the Apple to lab test equipment like function generators or plotters.

And we have tools to connect the Apple to the outside world, including A/D converters and interval timers with external interface.

We make components for the S-100 bus, the PET, and the TRS-80, too. We built our products to deliver hard-nosed value to the OEM, and to the inventor who knows the best, at prices that are unbeaten.

To find out how much computer your Apple II can be, see things our way. Because for serious users with serious uses for the Apple, we've got the tools.

California Computer Systems
250 Caribbean Sunnyvale, CA 94086 (408) 734-5811

A Micro for the Eighties

That's the author's assessment of the AlphaMicro AM-100 microcomputer.

Wm. C. Welborn, Jr.
112 N.W. 7th St.
Evansville IN 47708

Longtime readers of *Kilobaud* may remember articles by Dick Wilcox in three of the first four issues describing a complex operating system for microcomputers. Dick put his many years of programming experience not only into those articles, but also into a micro for the eighties, the AlphaMicro AM-100. As one who waited many months for this machine to become a reality, I can testify that Dick and the others at AlphaMicro have done a good job. Anyone buying this microcomputer will not be disappointed!

Introduction

The AM-100 hardware, while quite different from the usual micro boards, is surpassed only by the excellent software supported by this system. The CPU is a Western Digital WD-16, which, with its supporting chips, is mounted on two boards that plug into the S-100 bus (see Photos 1 and 2). Although the computer is a 16-bit machine, the CPU functions with 8-bit memories and I/O boards.

As a result, you don't have to spend double or triple the going microcomputer cost for your peripherals—almost all will work. You can upgrade your 8080 or Z-80 machine to the AM-100 for slightly more than the cost of the CPU boards. If you are just starting out, there are scores of devices that will work with the AM-100 and are totally compatible with it.

The CPU has a push-down hardware stack, vectored interrupt handling, eight 16-bit registers, hardware floating-point arithmetic and eight modes of addressing. But what really sets the AM-100 apart from the rest is the systems software that comes with it.

Stucture of the System

The standard AM-100 system, as supplied, is multitasking, multi-user, which means that several terminals can be operating at one time, handling different functions. This feature makes the AM-100 ideal for schools and for businesses that want multiple terminals on line, as well as for the personal computer user.

Each user has his own block of memory, up to 64K (not total, for each), and runs in a job area different from the other users' on the system. I have five separate jobs set up on my machine, for example. The system supervisor (that's you) can allocate jobs in just about whatever manner he wishes.

Areas of the disks are set up by the supervisor from time to time for different purposes, such as games, demonstration programs and business programs. There is no limit to the number of areas—called PPNs (Project-Programmer Number)—that you can allocate. (There is, however, a limit of 64 PPNs per disk.) Any PPN may be protected by a password to prevent unauthorized access. You don't have to allocate the number of disk blocks you want to use for each PPN; the system allocates blocks automatically as they are needed.

To keep track of all jobs, a system status display (DYSTAT) operates at all times through a video display monitor (VDM), which shows the supervisor what is going on with the system (see Photo 3). The VDM is supplied by you, but the software for DYSTAT is part of the system. (The VDM has been discontinued. DYSTAT will, I am told, work with other available video boards.)

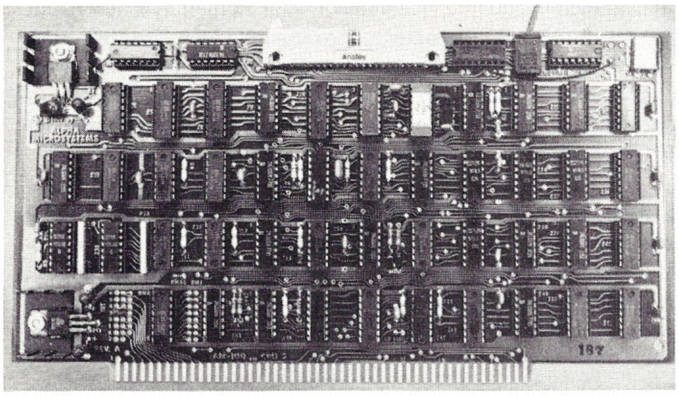
All system (and user) software is totally relocatable; that is, it will run anywhere in memory. This restricts writing assembly-language programs, as you cannot use any direct addressing modes. However, there are several addressing modes (not available for the 8080) that lend themselves to writing totally relocatable code.

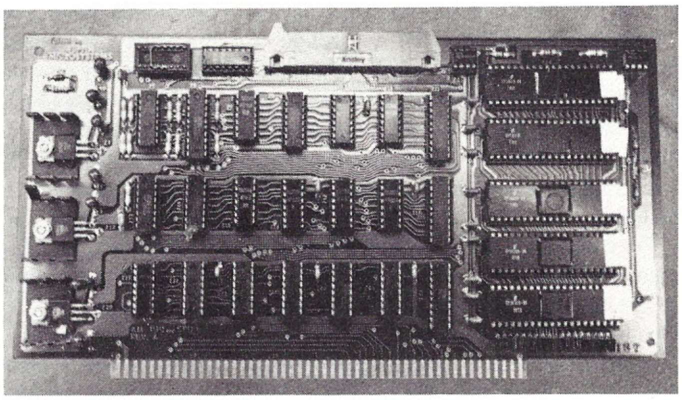
The system directs where in memory a program will end up without intervention by the operator. The system also keeps track of where your programs and data files are located on your disks. Photos 4 and 5 suggest how easy it is to use this system.

Terminal Service System

You can interface just about any terminal to the AM-100 and use just about any I/O board. AlphaMicro furnishes drivers for many popular boards and terminals, but if they don't support the ones you have, you can attempt to write a driver yourself.

Basically, you set up a terminal definition block one time for each terminal, printer or modem board that you intend to use. This block contains a reference to the port number of the I/O board you are using, the name of the driver (short assembly-language program that initializes that board, transfers characters to and from the system and looks for a data terminal ready (DTR) signal) and the make of terminal you have attached to the I/O board. The block also contains information about the size of the input and output buffers that you want the system to allocate for communication with that terminal.





Photos 1 and 2. The AM-100 CPU boards. The connector at the top of each board takes a ribbon cable, which interconnects the two boards. The large switch on the first board turns off the AM-100 and allows you to use your Z-80 or 8080 without removing the AM-100 from the box.

The terminal definition block (TRMDEF) is automatically put into the system when it is booted. Thereafter, a complex, but general, terminal service system scans each terminal in the system on an interrupt schedule, taking input from each in its time-slot (or outputting to each) so that the operation is transparent to the users. This is how the big time-sharing systems work and results in a satisfactory, flexible arrangement.

Systems Programs

There are several disk areas (PPNs) that are permanently reserved for systems programs and drivers. These come with the AM-100. On my system there are 143 programs in the system common area (accessible by all users), 46 device drivers (for terminals, printers, disks and I/O boards), eight demonstration programs in the LISP and Pascal languages and ten subroutines written in assembly language that are callable from BASIC. In addition, there are over 50 source listings for programs and drivers. Also included are 80-odd library routines, callable by assembly-language programs (mostly having to do with I/O functions), a raft of programs for use with the DC Hayes modem board and assorted BASIC source listings.

The latest update for the system (version 4.3) came out in January 1980. The documentation for the update alone was about an inch thick . . . in addition to an enormous amount of previous material.

While some programs are trivial—a 160-byte program that sets or gives you the time of day and an 18-byte program that rings the bell on your terminal—and are not too difficult to write yourself, most of the programs are substantial:

BASIC language program—over 10,500 bytes, plus 10K for the compiler and 10K for the package that runs the compiled object code.

ISAM files—to create and maintain indexed sequential files, usable either with BASIC or assembly-language programs.

EDIT—character-oriented editor that is similar to, although more comprehensive and easier to use than, the C/PN editor. I used this editor for creating my law-office system of programs and became quite familiar with its operation over a six-month period.

VUE—screen-oriented editor that will do more than EDIT and is faster and easier to use. VUE uses the cursor controls and simple 'CNTL-' combinations to accomplish its many functions. I have been using it extensively since its release last year and am totally sold on it. The biggest advantage is that text editing appears before you on the screen. To insert a line just hit CNTL-B and one will appear. Then type in the material you want there. You can either type over old text or toggle (with CNTL-Q) the insert mode; the old text will expand as you type in the new material or contract as you take out unwanted data. VUE even has some rudimentary text-formatting capability. There are over 50 functions.

TXTFMT—does extensive formatting of text: right margin justification, paragraphing, page length, title at top of each page if desired, page numbering top or bottom in a host of number types, vertical spacing, margin, sub-paragraph indentation and lettering. It will automatically create an index for you and output it at the end of your paper, if you wish. It will center text anywhere and print in boldface on command. I prepared this article with VUE and TXTFMT.

A macro assembler, a Dynamic Debugging Tool (DDT) program, Pascal and LISP are also supplied with the system.

Many of the programs supplied with the AM-100 system are used in the operation of the system itself. For example, LOG is used by the operator to get into a specific PPN and also performs other tasks; MEMORY allocates a block of memory for your use; ATTACH attaches your terminal to another job in the system; JOBPRI sets a priority for your job (i.e., a longer or shorter time-slice) compared to other jobs; MOUNT tells the system that you have changed disks.

Another group of programs are system utilities, which are unnecessary for operation of the system, but they make it easier to use. There are two fast sort utilities—one for use in BASIC programs and one for general use. A flexible copy utility, which will copy one program, one group of programs or an entire disk, is provided. Other utilities provide for erasing programs from disk; changing passwords and PPNs; dumping files in ASCII, hex or octal from disks to your terminal; appending one program to the tail end of another; and listing the files and programs in your disk area (or others) on your terminal. There are scores of such programs.

There is also a line-printer spooler to make it easy to queue up programs or files for printing. The spooler runs in its own job, and so, printing a program listing or data file does not monopolize the computer's time—you start it going and proceed to other tasks.

Command Language Processing

The AM-100 operating system supports a convenient and valuable tool called command language processing in which each entry from a terminal—when the system is in executive mode—is treated as a command by the system. A prompt character (period) indicates you are in this mode.

If you type DIR, the system will treat this as a command to look for a program of that name in the library section of the system's disk, load it into memory, execute it, output any information it has for you, delete it from memory and return you to executive mode. All this is done under program control.

Many commands require additional information. For instance, VUE requires the name of the program you wish to edit. LOG, if nothing else is given it, will tell you what disk area you are in; if a PPN follows the word LOG, you will be transferred to that PPN.

You can create a special file, called a command file, to take full advantage of command processing. A command file consists of a list of different commands you want executed. You call the file by typing its name. The system does the rest, executing each command, in order, until the file is exhausted.

For example, suppose you made a command file called BACKUP. (The actual name would be BACKUP.CMD. All programs and files have a six-letter (or less) name plus a three-letter extension. Often you can ignore the extension because the system recognizes "default" extensions. For example, the system will first look for a program called BACKUP. PRG (the extension for assembly-language object code modules). If the system can't locate this program, it will look for BACKUP.CMD, which it will load and run. On a two-terminal system with 20 megabytes of hard disk storage, the actual look-up time—from entry of a nonexistent program to error-message display—is less than 1.8 seconds; locating a command file takes less than .4 seconds.) In this file, store the commands that you would otherwise manually type in to make a backup of data files. As an added touch, use a few simple delimiters (:, < and >) on each side of any text you want to especially notice when you run the command file. While you're building this file, insert another delimiter (:K), which will stop the execution of the file until you hit return.

Now, all you have to do is type BACKUP, and the AM-100 system will automatically do all the rest—except physically load the disk into the driver. It will instruct you to do that and wait for you to hit return.

A command file called SYSTEM.INI is used to boot the system. This file contains the terminal definition blocks, definitions of the devices attached to the system, the line-printer-spooler parameters and other information essential to operate the system. The ROM-based bootstrap program, after start-up, looks for SYSTEM.INI on your system disk, loads it and then executes it. Thus, no matter how complex your system is, no matter how many printers, disks or terminals it has, all you have to do to operate it is to hit the reset button. Full documentation on how to write SYSTEM.INI is provided by AlphaMicro, and your dealer will provide the initial program to get your system running.

Whenever you log into a new disk area (PPN), the LOG program will look for a special command file. If one is there, LOG will read it and execute whatever commands are in the file. Typically, this could be used to run a compiled BASIC program for a business user or for school students.

AlphaBasic

The AlphaMicro crew has improved upon the BASIC language by implementing the following features:

1. Variables may be designated with any length name. You are not limited to just a letter or two. For example, you can call a variable representing the customer's last name CUSTLN, and not just C$ or CN$. In fact, you could call it CUSTOMER'LAST' NAME if you had plenty of memory to spare for such nonsense and didn't mind typing that long, awkward form wherever it appeared in the program. All letters, not just the first two, are significant.

2. Memory mapping. For those of you who speak COBOL, this is similar to the data-formatting capabilities of that language. One difference is that, in COBOL, the variables must be defined in a separate section of the program, whereas in AlphaBasic, they may but need not be. You can use this feature in I/O techniques. For example, it is convenient to read many variables from a disk record (or write to disk) by using one key variable to refer to all the rest. In the programs I wrote for our law office, for instance, there are perhaps 30 variables in the client master record. I mapped these variables in AlphaBasic and refer to the whole group as INREC (INformation RECord).

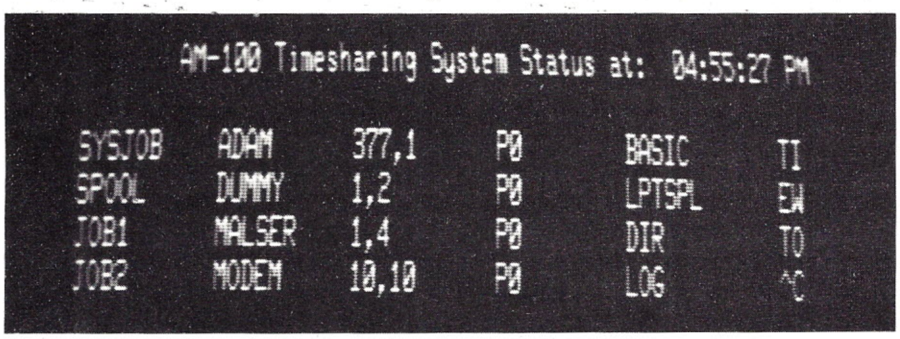


Photo 3. DYSTAT display is on a monitor apart from any terminal and runs all the time. The display lists the jobs (column 1), the terminal (col. 2), disk area (col. 3), priority (col. 4), program (col. 5) and condition (col. 6).

Whenever I want to read in a new client master record from the disk, the statement "READ #1, INREC" will access the entire record and read it into memory. (If I want to refer to any one of the 30-odd components of that record, I am free to do so.)

Memory mapping is unique to AlphaBasic and results in a language that is superior to any other BASIC in operation and programming. The departure from "standard" is well worth the extra flexibility of the language. Other uses for this feature will become apparent as you use it.

3. Labeled subroutines. This makes line numbers almost obsolete. If you have ever spent hours of frustrating debugging time over a long program, only to discover that the only problem is a call to a botched-up line number, you will appreciate this feature. Moreover, it makes BASIC programs more readable. Instead of a call to, say, line 2500, where a subroutine is to be found, you code it this way: CALL DOIT. Then label line 2500 like this: 2500 DOIT:. You are not limited to just subroutines; GOTOs are also valid when followed by a label. Essentially, the BASIC programmer is thus freed from keeping track of any line numbers. You can dispense with them entirely, in fact, by entering your BASIC program with the editor instead of the usual way. I haven't tried this yet; anyway, the compiler generates error messages referenced to line numbers.

Space simply does not permit a complete discussion of AlphaBasic, which includes features not found in any, or at least in few, other versions of BASIC. Bear in mind that the three unique features mentioned above are in addition to the many improvements in the original BASIC that have been made by others.

AlphaBasic may be used in an immediate mode, just like most of the other versions. Type "PRINT 2+2" (while in BASIC and without a line number), and the system will print a figure "4." It may be used in what we have come to think of as "normal," as in the interpreter versions (type in your program and say RUN; it will).

However, AlphaBasic is also a compiled language. You can compile the BASIC program and save the object code on disk. Thereafter, you needn't enter BASIC at all. Among other things, this means that you, or a command file, may run BASIC programs from executive mode. Furthermore, using the compiled code results in a great speed advantage. Also, the space occupied by the compiled program is far less than the source listing of BASIC statements. (For example, my Lawyer Billing source code is 30,720 bytes, while the compiled code is only 9202 bytes!) Compiled code permits you to market applications packages in BASIC, without revealing the source code. The system also permits you to compile saved BA-

SIC programs without entering BASIC. So, you don't have to worry about trying to retain sufficient memory for the compiled code while entering your source code.

Critical Comments

One of the most troublesome problems was lack of documentation. The AM-100 is supported by a complex operating system that demands full explanation. The original documents concerning its use left a lot to be desired. There are still some deficiencies in this area; however, the last four releases (versions 4.0 to 4.3) of the system not only were well documented with regard to the new programs and latest changes, but also contained additional information on the use of the system's original features. In other words, AlphaMicro is doing something about this problem. As usual, they are doing it well. It just took a little while for them to get around to it!

There are still some bugs in the system. Most of them are minor. Known bugs are documented by AlphaMicro. That policy is a change from some software suppliers who treat the bugs in their programs as if they were holy: never to be discussed with the unwashed users. Furthermore, the AlphaMicro people want to hear about the bugs you find. They supply convenient forms to make it easy for you to report them.

The final problems concern memory. First—and this really is not criticism, but a fact of life—the system uses a lot of memory. Don't think you can buy an AM-100, put it in a box with 24K and then upgrade later. You can't, with any degree of satisfaction. Figure on filling the box with at least 64K.

Second, the system will not function with just any memory boards. This is without question the most critical hardware requirement. I don't know the cause, but if your memory chips are the least bit flaky, you are going to have big problems. And, you can't use most dynamic memory boards at all. If you have any problems with the system, chances are they are caused by "bad" memory (which may function without hitch with another CPU board).

Cost

No one ever said the AM-100 was cheap! The two CPU boards are $1495, which includes the software license for the system software, but no disks with programs on them. The disks come with the disk controller board. But there's a catch-22. You can't use just any disk controller. Generally speaking, you will have to buy a controller (for floppies or hard disk) from AlphaMicro—even if you already have the drive itself—or the programs you are getting just won't work. This will mean, at a minimum, spending another $695, for a total of $2190.

This may seem like a lot, but if you calculate the total price of a CPU, hardware floating-point arithmetic board, disk controller, a quality operating system, an editor, assembler, word-processing package and miscellaneous utility software, you will come up with a surprising total . . . without the features of the AM-100!

Also consider that any business system is going to have disk drives, terminals and memory that total far more than $2000. In other words, those CPU boards are not a large percentage of the total cost of a complete system.

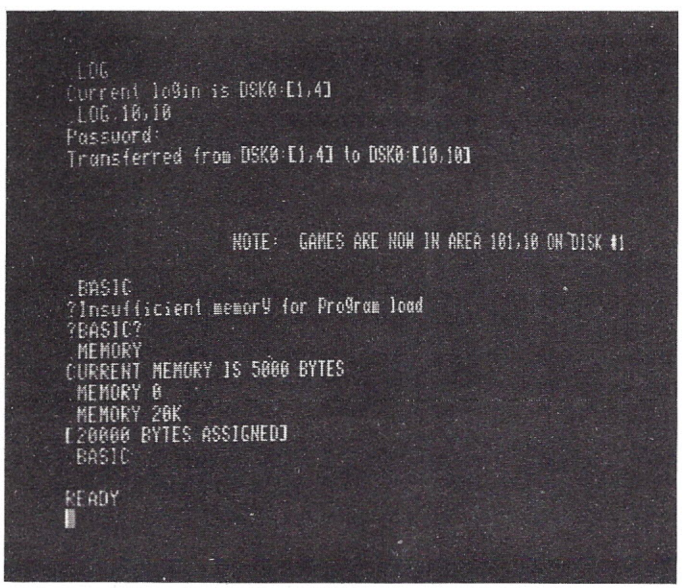
The AM-100 is not inexpensive; however, for the price, you would be hard pressed to find a computer with as complex an operating system, a better language for business programs or the capacity to handle multiple terminals. Admitting I am prejudiced, but having spent months surveying the market before making a decision to buy, I think the AlphaMicro AM-100 is the best you can buy for less than $40,000 or $50,000. The AM-100, with everything you could hang on it, will cost about a third of that—lots less if you stay with floppy disks and shop around for a used line printer. (A hard disk and a new line printer can easily cost over $8000.)

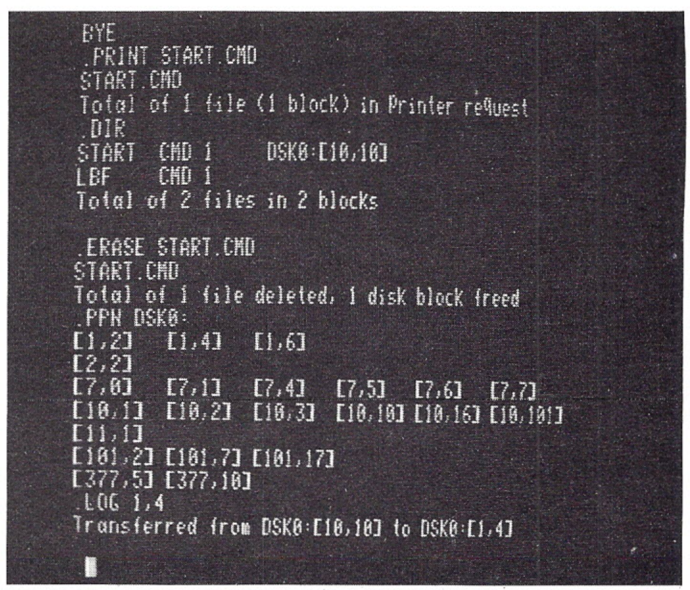
Applications Software

AlphaMicro has a fine business accounting package for $750. This price, considering the quality of the programs, is inexpensive, but it is also a bit misleading. You have to tailor the package to your own business. If you don't program in BASIC yourself, you will have to spend some bucks (maybe $1000 or more) getting this work done for you.

There is a growing supply of other applications software designed for the AM-100. The AlphaMicro Users' Society (AMUS), based in Colorado, has a catalog of all the software presently available for the AM-100 from various suppliers.

Many AlphaMicro dealers have software of their own. The company picks its dealers with care. As a result, most are capable of generating not only good service and a friendly concern, but also good applications programs. My dealer (Data Domain in Bloomington, Indiana) not only helped me put together a workable system for my needs, but also assisted me in working out various problems I have had (most of which were the result of my own ignorance). This help was given freely, in spite of the fact





Photos 4 and 5. Operations frequently used with the AM-100. The system prompt symbol (period) indicates operator input; other lines were typed by the computer.

that I was writing all of my own applications programs. They did write a driver for my Malibu printer (not supported, yet, by AlphaMicro) . . . at no charge, I might add.

Software for the IDS Modem-88, as well as other software, is available from Khalsa Computer Systems of Pasadena CA.

Dick Wilcox and the rest of the AlphaMicro crew have gone out of their way to ensure quality and competence on the part of their dealers, as well as with their system. This means that you, as an end user, can probably look to your dealer for whatever assistance you need in the way of applications programs. Contact me if you are interested in a Lawyer Billing File package—mostly for accounts receivable.

Summary

The AM-100 system is truly a microcomputer system for the eighties. I doubt greatly that any company will equal this system, let alone surpass it, for many years to come. It is superior to many minicomputers that I have personally checked out—computers costing five to ten times as much. The manufacturer is AlphaMicro, 17881 Sky Park North, Irvine CA 92714.

I have an AM-100 system that has been operating for over a year, causing no trouble at all, doing what I want of it and making me money. What more can be said? ■

• PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET

PET PRODUCTS

Programs — Workbooks
for Floppy Disk — for Cassette

See your Dealer

Dealer Inquiries Invited

Put your PET to work!

T I S

PROGRAMS

- **SW-1*** **MAILB** mailing list system
- **SW-2*** **CHECKBOOK** record
- **SW-3*** **ACCOUNTS** keep track of who owes you how much
- **SW-4** **MEDIT** create and maintain data files
- **SW-5*** **CALENDAR** appointments, meetings at a glance

WORKBOOKS

WB-1	**Getting Started with Your PET**	$3.95
WB-2	**PET String and Array Handling**	$3.95
WB-3	**PET Graphics**	$4.95
WB-4	**PET Cassette I/O**	$4.95
WB-5	**Miscellaneous PET Features**	$3.95
WB-6	**PET Control and Logic**	$3.95

*These programs are special purpose data base management systems. They all can:

- Sort numeric or string fields
- Select based on numeric or string (=)
- Select based on substring match
- Select based on range of entry number

Prices: $9.90 each for programs using cassette storage for data
$12.95 each using sequential floppy disk storage for data.
Price includes 40 - 60 page instruction manual

Add $2.00 for shipping and handling

Money back guarantee

TIS
P.O. Box 921, Dept. KB
Los Alamos, NM 87544

VISA

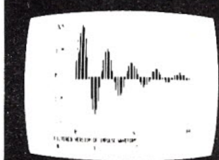
On bank card orders, give all numbers

✓95

PET is a trademark of Commodore Business Machines

• PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET

TRS-80



- *F(AST) F(OURIER) T(RANSFORM)*
- *DIGITAL FILTER SIMULATION*
- *LINEAR AND EXPONENTIAL CURVE FIT*
- *DISK OR CASSETTE DATA FILES*
- *INTERACTIVE GRAPHICS*

- Having this set of interactive programs in your hands is a learning experience in digital signal processing.
- Learn by doing. Documentation includes multiple examples. Balance your checkbook with a digital filter (can you believe it?). Plot daily stock market values and their computed trend lines. Find the frequency response of a digital filter. Illustrate nyquist sampling theorem. Perform spectral analysis (FFT) on any waveform.

TRS-80

- *IDEAL FOR YOUR PHILATELIC INVENTORY*
- *FOR MEDIUM TO LARGE COLLECTIONS*
- *MULTIPLE PARAMETER SORTING*
- *VERSATILE PRINTING CAPABILITY*
- *OPTIMIZED FOR FAST INITIAL ENTRY*

- One thousand stamps per Data disk with Multi-disk processing for the large collection.
- Special attention given to make data entry and corrections fast and easy. Keeps your information current and accurate.
- Provides a variety of pre-designed and user-selected reports to CRT or printer. Reports are sorted on any combination of stamp characteristics.
- Requires minimum Dual disk 32K system and 72+ column printer.

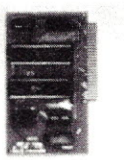
DELTA SYSTEMS

VISA

P.O. BOX 1181
GOLETA, CA
93017

FFT-80 DISK	**$30.00**
FFT-80 CASSETTE	**$25.00**
STAMPS-80 DISK	**$35.00**

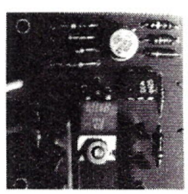
APPLE II PARALLEL INTERFACE CARD



John Bell Engineering is announcing an Apple II Parallel Interface Card. There are four I/O ports with handshaking logic. The board has two 6522 versatile interface adapters and a 74LS74 for addressing and timing. Each 6522 has two interval timers. This will interface your Apple II to printers, speech synthesizers, keyboards, and other John Bell Engineering products. Inputs and outputs are TTL and CMOS compatible.

Prices:
79-295 Complete kit $69.95
79-295 Assembled $79.95

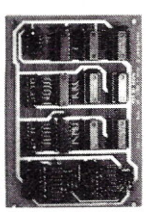
SOLID STATE SWITCH



Now you can control the World! Switch lights on and off for home security, computer controlled disco light shows. Turn your printer on only when needed. The Switch can handle 720 watts (120 VAC 6 AMPS). Its input is TTL compatible (5V-2MA), isolation 1500VDC. The circuit board is 2" square on the one channel kit and 2"x 8" on the 4 channel unit.

Prices:
1 Channel kit $9.95 assm. $12.50
4 Channel kit $34.95 assm. $44.95

A to D D to A CONVERTER



John Bell Engineering now has available an Analog to Digital and Digital to Analog Converter Board. Featues low cost medium speed (50,000 conversions per second) for applications such as speech recording and music synthesizing. Single power (+5) required. Parallel inputs and outputs include 8 data bits, strobe lines, and latches. Analog inputs and outputs are medium impedance zero to five volt range.

Prices:
79-287 Kit $49.95
79-287 Assembled $69.95

PRODUCTS AVAILABLE FROM:

JOHN BELL ENGINEERING ✓99
P.O. BOX 338
DEPT. 4
REDWOOD CITY, CA 94064
(415) 367-1137

ADD 6% SALES TAX IN CALIFORNIA AND $1.00 SHIPPING & HANDLING FOR ORDERS LESS THAN $20. ADD 4% FOR VISA OR M.C.

J B ENGINEERING

DISK SORT MERGE 'DSM' FOR MOD I AND MOD II TRS-80™

FAST → **Now you can sort an 85K diskette in less than 3 minutes*** **← FAST**

Perfect for your multi-diskette RANDOM file mailing lists, inventory, etc. Ideal for specialized report generation. Sort, merge or combination. All machine language stand-alone package — Efficient and easy to use. No separate key files required! Physical records are rearranged on diskette! Supports multiple sub records per sector including optional sector spanning. Sorts on one or more fields — ascending or descending. Sort fields within records may be character, integer, and floating-point binary. Provides optional output field deletion, rearrangement, and padding.

*Sort timings shown below are nominal times. Times will vary based on sort and system configurations. Nominal times based on Mod I 48K 4-drive configuration, 64 byte records, and 5 sort keys.

TYPE	FILE SIZE (Bytes)	SORT TIME (Sec)	TYPE	FILE SIZE (Bytes)	SORT TIME (Sec)
SORT	16K	33	SORT	340K	1081
SORT	32K	49	SORT	680K	2569
SORT	85K	173	SORT and MERGE	85K SORT + 1275K Merge	1757
SORT	170K	445			

DSM for Mod I (Minimum 32K, 2-drives) **$75 On-Disk**
DSM for Mod II (Minimum 64K, 1-drive) **$150 On-Disk**
Mod II Development Package $100
Machine Language SUPERZAP, plus Editor/Assembler and Disassembler patches. (Include copy of Apparat NEWDOS + 5¼ diskette.)
Mod II Generalized Subroutine Facility 'GSF' $50
∞ **BASIC** for Level II and Disk Systems **$49.95**
∞ **BUSINESS** (Requires Infinite BASIC) **$29.95**
COMPROC Command Processor for Disk Systems **$19.95**
REMODEL + PROLOAD (Specify 16, 32, or 48K Memory) **$34.95**
GSF (Specify 16, 32, or 48K) **$24.95**

CHECK, VISA, M/C, C.O.D.
Calif. Residents add 6%
Telephone Orders Accepted
(714) 637-5016

✓101

RACET COMPUTES
702 Palmdale, Orange CA 92665

TRS-80 IS A REGISTERED TRADEMARK OF TANDY CORPORATION

NEW! TPM* for TRS-80 Model II
NEW! System/6 Package
Computer Design Labs

Z80* Disk Software

We have acquired the rights to all TDL software (& hardware). TDL software has long had the reputation of being the best in the industry. Computer Design Labs will continue to maintain, evolve and add to this superior line of quality software.
— Carl Galletti and Roger Amidon, owners.

Software with Manual/Manual Alone

All of the software below is available on any of the following media for operation with a Z80 CPU using the CP/M* or similar type disk operating system (such as our own TPM*).

for TRS-80* CP/M (Model I or II)
for 8" CP/M (soft sectored single density)
for 5¼" CP/M (soft sectored single density)
for 5¼" North Star CP/M (single density)
for 5¼" North Star CP/M (double density)

BASIC I

A powerful and fast Z80 Basic interpreter with EDIT, RENUMBER, TRACE, PRINT USING, assembly language subroutine CALL, LOADGO for "chaining", COPY to move text, EXCHANGE, KILL, LINE INPUT, error intercept, sequential file handling in both ASCII and binary formats, and much, much more. It runs in a little over 12 K. An excellent choice for games since the precision was limited to 7 digits in order to make it one of the fastest around. $49.95/$15.

BASIC II

Basic I but with 12 digit precision to make its power available to the business world with only a slight sacrifice in speed. Still runs faster than most other Basics (even those with much less precision). $99.95/$15.

BUSINESS BASIC

The most powerful Basic for business applications. It adds to Basic II with random or sequential disk files in either fixed or variable record lengths, simultaneous access to multiple disk files, PRIVACY command to prohibit user access to source code, global editing, added math functions, and disk file maintenance capability without leaving Basic (list, rename, or delete). $179.95/$25.

ZEDIT

A character oriented text editor with 26 commands and "macro" capability for stringing multiple commands together. Included are a complete array of character move, add, delete, and display function. $49.95./$15.

ZTEL

Z80 Text Editing Language - Not just a text editor. Actually a language which allows you to edit text and also write, save, and recall programs which manipulate text. Commands include conditional branching, subroutine calls, iteration, block move, expression evaluation, and much more. Contains 36 value registers and 10 text registers. Be creative! Manipulate text with commands you write using Ztel. $79.95/$25.

TOP

A Z80 Text Output Processor which will do text formatting for manuals, documents, and other word processing jobs. Works with any text editor. Does justification, page numbering and headings, spacing, centering, and much more! $79.95/$25.

MACRO I

A macro assembler which will generate relocateable or absolute code for the 8080 or Z80 using standard Intel mnemonics plus TDL/Z80 extensions. Functions include 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical operations, local and global symbols, chaining files, linking capability with optional linker, and recursive/reiterative macros. This assembler is so powerful you'll think it is doing all the work for you. It actually makes assembly language programming much less of an effort and more creative. $79.95/$20.

MACRO II

Expands upon Macro I's linking capability (which is useful but somewhat limited) thereby being able to take full advantage of the optional Linker. Also a time and date function has been added and the listing capability improved. $99.95/$25.

LINKER

How many times have you written the same subroutine in each new program? Top notch professional programmers compile a library of these subroutines and use a Linker to tie them together at assembly time. Development time is thus drastically reduced and becomes comparable to writing in a high level language but with all the speed of assembly language. So, get the new CDL Linker and start writing programs in a fraction of the time it took before. Linker is compatible with Macro I & II as well as TDL/Xitan assemblers version 2.0 or later. $79.95/$20.

DEBUG I

Many programmers give up on writing in assembly language even though they know their programs would be faster and more powerful. To them assembly language seems difficult to understand and follow, as well as being a nightmare to debug. Well, not with proper tools like Debug I. With Debug I you can easily follow the flow of any Z80 or 8080 program. Trace the program one step at a time or 10 steps or whatever you like. At each step you will be able to see the instruction executed and what it did. If desired, modifications can then be made before continuing. It's all under your control. You can even skip displaying a subroutine call and up to seven breakpoints can be set during execution. Use of Debug I can pay for itself many times over by saving you valuable debugging time. $79.95/$20.

DEBUG II

This is an expanded debugger which has all of the features of Debug I plus many more. You can "trap" (i.e. trace a program until a set of register, flag, and/or memory conditions occur). Also, instructions may be entered and executed immediately. This makes it easy to learn new instructions by examining registers/memory before and after. And a RADIX function allows changing between ASCII, binary, decimal, hex, octal, signed decimal, or split octal. All these features and more add up to give you a very powerful development tool. Both Debug I and II must run on a Z80 but will debug both Z80 and 8080 code. $99.95/$20.

ZAPPLE

A Z80 executive and debug monitor. Capable of search, ASCII put and display, read and write to I/O ports, hex math, breakpoint, execute, move, fill, display, read and write in Intel or binary format tape, and more! on disk

APPLE

8080 version of Zapple

NEW! TPM now available for TRS-80 Model II!

TPM*

A NEW Z80 disk operation system! This is not CP/M*. It's better! You can still run any program which runs with CP/M* but unlike CP/M* this operating system was written specifically for the Z80* and takes full advantage of its extra powerful instruction set. In other words its not warmed over 8080 code! Available for TRS-80* (Model I or II). Tarbell, Xitan DDDC, SD Sales "VERSA-FLOPPY", North Star (SD&DD), and Digital (Micro) Systems. $79.95/$25.

SYSTEM MONITOR BOARD (SMB II)

A complete I/O board for S-100 systems. 2 serial ports, 2 parallel ports, 1200/2400 baud cassette tape interface, sockets for 2K of RAM, 3-2708/2716 EPROM's or ROM, jump on reset circuitry. Bare board $49.95/$20.

ROM FOR SMB II

2KX8 masked ROM of Zapple monitor. Includes source listing $34.95/$15.

PAYROLL (source code only)

The Osborne package. Requires C Basic 2.
5" disks $124.95 (manual not included)
8" disks $ 99.95 (manual not included)
Manual $20.00

ACCOUNTS PAYABLE/RECEIVABLE (source code only)

By Osborne, Requires C Basic 2
5" disks $124.95 (manual not included)
8" $99.95 (manual not included)
Manual $20.00

GENERAL LEDGER (source code only)

By Osborne. Requires C Basic 2
5" disks $99.95 (manual not included)
8" disks $99.95 (manual not included)
Manual $20.00

C BASIC 2

Required for Osborne software. $99.95/$20.

SYSTEM/6

TPM with utilities, Basic I interpreter, Basic E compiler, Macro I assembler, Debug I debugger, and ZEDIT text editor.
Above purchased separately costs $339.75
Special introductory offer: Only $179.75 with coupon!!

$160.

This Coupon Is Worth One Hundred And Sixty Dollars Toward The Full Price Of The SYSTEM/6 Package
System/6 with this coupon is only $179.95.
This is a limited time offer.

$160.00

ORDERING INFORMATION

Visa, Master Charge and C.O.D. O.K. To order call or write with the following information.

1. Name of Product (e.g. Macro I)
2. Media (e.g. 8" CP/M)
3. Price and method of payment (e.g. C.O.D.) include credit card info. if applicable.
4. Name, Address and Phone number.
5. For TPM orders only: Indicate if for TRS 80, Tarbell, Xitan DDDC, SD Sales (5¼" or 8"). ICOM (5¼" or 8"), North Star (single or double density) or Digital (Micro) Systems.
6. N.J. residents add 5% sales tax.

Manual cost applicable against price of subsequent software purchase in any item except for the Osborne software.

For information and tech queries call
609-599-2146

For phone orders ONLY call toll free
1-800-327-9191
Ext. 676
(Except Florida)

OEMS

Many CDL products are available for licensing to OEMs. Write to Carl Galletti with your requirements.

* Z80 is a trademark of Zilog
* TRS-80 is a trademark for Radio Shack
* TPM is a trademark of Computer Design Labs. It is not CP/M*
* CP/M is a trademark of Digital Research

Prices and specifications subject to change without notice.

DEALER INQUIRIES INVITED.

COMPUTER DESIGN LABS
342 Columbus Avenue
Trenton, N.J. 08629

✓18

✓Reader Service—see page 258

Buy or Rent?

```
RUN
              ***BUY OR RENT ? ***


ENTER PRICE OF HOUSE---$ 41100

ENTER AMOUNT OF 'FRONT-MONEY' TO BE APPLIED TO
PURCHASE.(TO PAY DOWN-PAYMENT AND CLOSING COSTS)---$1650

ENTER ESTIMATED CLOSING COSTS (ATTORNEY'S FEE ,
''POINTS'',etc.---$1650

ENTER EST. YEARLY PROP. TAXES AND INSURANCE---$450

ENTER INTEREST RATE OF MORTGAGE---% 9.5

ENTER LENGTH OF MORTGAGE IN MONTHS 360

ENTER THE NUMBER OF MONTHS YOU WILL LIVE IN THE HOUSE/APT 39

ENTER YOUR ESTIMATED AVERAGE MARGINAL TAX RATE FOR
THE PERIOD YOU WILL LIVE IN THE HOUSE/APT---%31

ENTER THE YEARLY APPRECIATION EXPECTED ON THE HOUSE---%9

WHEN YOU SELL,WILL YOU USE A REAL ESTATE AGENT? (Y OR N) Y

ENTER THE %-AGE OF THE CUSTOMARY AGENT'S COMMISSION---% 6

ENTER AMOUNT OF MONTHLY RENT---$ 210

ENTER ESTIMATED YEARLY INCREASE IN RENT---% 6

ENTER THE BEST INVESTMENT INTEREST AVAILABLE TO YOU
OVER THE LENGTH OF YOUR STAY (COMP.QTRLY)---% 6.5

IF YOU RENT WILL YOU INVEST THE DIFFERENCE BETWEEN THE MORTGAGE
AND RENT PAYMENTS ? (Y OR N)Y


                    COMPUTING


                 ***BUYING INFO***
PRICE OF HOUSE                                   $41,100.00
'FRONT-MONEY'                                     $1,650.00
CLOSING COSTS                                     $1,650.00
MONTHLY PROP. TAXES AND INSURANCE                    $37.50
AMOUNT OF MORTGAGE                               $41,100.00
INTEREST RATE ON MORTGAGE                            9.50 %
MONTHLY PAYMENT (INCLUDES TAXES & INS.)             $383.09
ENTER 'A' FOR COST ANALYSIS.A
                  **COST ANALYSIS**
VALUE OF HOUSE AT END OF STAY                    $55,004.59
MORTGAGE BAL. AT END OF STAY                   ( $40,180.56)
AGENT'S COMMISSION                             (  $3,300.28)
NET PROCEEDS FROM SALE                           $11,523.75
                 *************
LENGTH OF STAY                                   39 MONTHS
TOTAL OF PAYMENTS MADE                           $16,590.55
TAX SAVINGS FROM BUYING                        (  $3,893.17)
NET PROCEEDS FROM SALE                         ( $11,523.75)
TOTAL COST OF BUYING                              $1,173.63
MONTHLY COST OF BUYING                               $30.09
ENTER 'R' TO DISPLAY RENTAL INFOR
                 ***RENTAL INFO***
INITIAL MONTHLY RENT                                $210.00
FINAL MONTHLY RENT                                  $250.11
TOTAL OF RENTAL PAYMENTS                          $8,773.01
INTEREST ON 'FRONT-MONEY' SAVED                     $126.84
INTEREST ON MTG-RENT DIFFERENCE INVESTED             $33.94
NET COST OF RENTING                               $8,612.23
MONTHLY COST OF RENTING                             $220.83
REVIEW INFO ? (Y OR N)N
READY
```

Sample run 1.

```
RUN
                 ***BUY OR RENT ? ***


ENTER PRICE OF HOUSE---$ 62000

ENTER AMOUNT OF 'FRONT-MONEY' TO BE APPLIED TO
PURCHASE.(TO PAY DOWN-PAYMENT AND CLOSING COSTS)---$14570

ENTER ESTIMATED CLOSING COSTS (ATTORNEY'S FEE ,
''POINTS'',etc.---$2170

ENTER EST. YEARLY PROP. TAXES AND INSURANCE---$820

ENTER INTEREST RATE OF MORTGAGE---% 10

ENTER LENGTH OF MORTGAGE IN MONTHS 360

ENTER THE NUMBER OF MONTHS YOU WILL LIVE IN THE HOUSE/APT 60

ENTER YOUR ESTIMATED AVERAGE MARGINAL TAX RATE FOR
THE PERIOD YOU WILL LIVE IN THE HOUSE/APT---%38

ENTER THE YEARLY APPRECIATION EXPECTED ON THE HOUSE---%9

WHEN YOU SELL,WILL YOU USE A REAL ESTATE AGENT? (Y OR N) Y

ENTER THE %-AGE OF THE CUSTOMARY AGENT'S COMMISSION---% 6

ENTER AMOUNT OF MONTHLY RENT---$ 300

ENTER ESTIMATED YEARLY INCREASE IN RENT---% 3

ENTER THE BEST INVESTMENT INTEREST AVAILABLE TO YOU
OVER THE LENGTH OF YOUR STAY (COMP.QTRLY)---% 8

IF YOU RENT WILL YOU INVEST THE DIFFERENCE BETWEEN THE MORTGAGE
AND RENT PAYMENTS ? (Y OR N)Y


                    COMPUTING


                 ***BUYING INFO***
PRICE OF HOUSE                                   $62,000.00
'FRONT-MONEY'                                    $14,570.00
CLOSING COSTS                                     $2,170.00
MONTHLY PROP. TAXES AND INSURANCE                    $68.33
AMOUNT OF MORTGAGE                               $49,600.00
INTEREST RATE ON MORTGAGE                           10.00 %
MONTHLY PAYMENT (INCLUDES TAXES & INS.)             $503.61
ENTER 'A' FOR COST ANALYSIS.A
                  **COST ANALYSIS**
VALUE OF HOUSE AT END OF STAY                    $97,072.22
MORTGAGE BAL. AT END OF STAY                   ( $47,900.81)
AGENT'S COMMISSION                             (  $5,824.33)
NET PROCEEDS FROM SALE                           $43,347.08
                 *************
LENGTH OF STAY                                   60 MONTHS
TOTAL OF PAYMENTS MADE                           $44,786.54
TAX SAVINGS FROM BUYING                        (  $9,278.60)
NET PROCEEDS FROM SALE                         ( $43,347.08)
TOTAL COST OF BUYING                             $-7,839.15
MONTHLY COST OF BUYING                             $-130.65
ENTER 'R' TO DISPLAY RENTAL INFOR
                 ***RENTAL INFO***
INITIAL MONTHLY RENT                                $300.00
FINAL MONTHLY RENT                                  $347.78
TOTAL OF RENTAL PAYMENTS                         $19,112.89
INTEREST ON 'FRONT-MONEY' SAVED                   $1,417.16
INTEREST ON MTG-RENT DIFFERENCE INVESTED             $79.36
NET COST OF RENTING                              $17,616.37
MONTHLY COST OF RENTING                             $293.61
REVIEW INFO ? (Y OR N)N
READY
```

Sample run 2.

Ed Pons
404 Lake Road
Havelock NC 28532

Every year the addresses of 20 percent of the American population change. Many of these transients are corporate executives or mid- to upper-level government employees whose career patterns require frequent transfers. A move may be the result of a job change or promotion with an increase in earnings.

In any event, you probably have a friend who is deciding whether to enter the home-buying market—either for the first time or for a limited stay. This program will be helpful to such a decision-maker, while providing the computer hobbyist another answer to *the question.*

The Program

The user first supplies information concerning the property he is considering buying, his tax rate, anticipated length of stay and return on investments. Rental data is also entered, and the user is asked whether, in the event he rents, he will invest the difference between his rent and what his house payment would have been.

It should be noted that the yearly appreciation on the house is compounded monthly. This was done so stays of less than twelve months could be evaluated.

The program was designed to help the newcomer to the housing market and does not take into account the recognition of gain to the taxpayer who sold a previous home at a profit and who does not reinvest in a house (Internal Revenue Code, sec. 1034). The user may wish to add a few lines to apply T1 (from line 150) to the amount of capital gain and add the result to the cost of renting.

In the printing of cost information, parentheses are used to identify amounts that are subtracted from an above unparenthesized amount in order to yield the amount appearing directly under the last parenthesized amount. Thus, in Sample run 1 under "Cost Analysis," $40,180.56 and $3,300.28 are subtracted from $55,004.59 to yield $11,523.75.

Negative numbers are preceded by a minus sign as in the "Cost of Buying" amounts in Sample run 2.

Sample run 1 illustrates a VA-"nothing down" loan, and Sample run 2 deals with a conventional loan with a 20 percent down payment. In both runs the user-supplied input assumes a low property tax rate found in a rural area.

Program listing.

```
LIST

  10 PRINT TAB(10),"***BUY OR RENT ? ***": PRINT : PRINT : PRINT
  20 REM--- PROGRAM BY ED PONS
  30 INPUT "ENTER PRICE OF HOUSE---$ ",P1: PRINT
  40 PRINT "ENTER AMOUNT OF 'FRONT-MONEY' TO BE APPLIED TO "
  50 INPUT "PURCHASE.(TO PAY DOWN-PAYMENT AND CLOSING COSTS)---$",D1
  60 PRINT
  70 PRINT "ENTER ESTIMATED CLOSING COSTS (ATTORNEY'S FEE ,"
  80 INPUT "''POINTS'',etc.---$",C1: PRINT
  90 INPUT "ENTER EST. YEARLY PROP. TAXES AND INSURANCE---$",T5: PRINT
 100 INPUT "ENTER INTEREST RATE OF MORTGAGE---% ",I1: PRINT
 110 INPUT "ENTER LENGTH OF MORTGAGE IN MONTHS ",L1: PRINT
 120 INPUT "ENTER THE NUMBER OF MONTHS YOU WILL LIVE IN THE HOUSE/APT ",S1
 130 PRINT
 140 PRINT "ENTER YOUR ESTIMATED AVERAGE MARGINAL TAX RATE FOR"
 150 INPUT "THE PERIOD YOU WILL LIVE IN THE HOUSE/APT---%",T1
 160 PRINT
 170 INPUT "ENTER THE YEARLY APPRECIATION EXPECTED ON THE HOUSE---%",A1
 180 PRINT
 190 INPUT "WHEN YOU SELL,WILL YOU USE A REAL ESTATE AGENT? (Y OR N) ",Z$
 200 PRINT
 210 LET B1=0
 220 IF Z$="N" THEN 250: IF Z$<>"Y" THEN 190
 230 INPUT "ENTER THE %-AGE OF THE CUSTOMARY AGENT'S COMMISSION---% ",B1
 240 PRINT
 250 INPUT "ENTER AMOUNT OF MONTHLY RENT---$ ",R1: PRINT
 260 INPUT "ENTER ESTIMATED YEARLY INCREASE IN RENT---% ",A2: PRINT
 270 PRINT "ENTER THE BEST INVESTMENT INTEREST AVAILABLE TO YOU"
 280 INPUT "OVER THE LENGTH OF YOUR STAY (COMP.QTRLY)---% ",I6: PRINT
 290 PRINT "IF YOU RENT WILL YOU INVEST THE DIFFERENCE BETWEEN THE MORTGAGE"
 300 INPUT "AND RENT PAYMENTS ? (Y OR N)",A$: PRINT
 310 IF A$="N" THEN 330
 320 IF A$<>"Y" THEN 290
 330 PRINT : PRINT TAB(9),"&D&D&D&D&D&D&D COMPUTING &D&D&D&D&D&D&D"
 340 PRINT : PRINT
 350 REM- MORTGAGE CALCULATIONS
 360 REM-AMT OF MTG
 370 LET P2=P1+C1-D1
 380 REM-AMT OF MONTHLY PMT (P3)
 390 LET I2=I1/1200
 400 LET P3=P2*(I2/(1-(1/(1+I2)↑L1)))
 410 LET T5=T5/12
 420 REM-CALC. CUM AMT PAID(P4);MTG BAL(B2);CUM INT(I4);
 430 REM- COST TO OWNER(C2);&TAX SAVINGS(P6)
 440 LET P4=0: LET B2=P2: LET I4=0: LET C2=0
 450 FOR K=1 TO S1
 460   LET P4=P4+P3+T5
 470   LET I3=B2*I2: LET I3=I3*100+.5: LET I3=INT(I3)/100
 480   LET P5=P3-I3: LET B2=B2-P5: LET I4=I4+I3
 490   LET C2=C2+P3+T5-I3*T1/100
 500 NEXT K
 510 LET P6=P4-C2
 520 REM-CALC RENT;CUM RENT(R2)
 530 LET R2=0: LET R3=R1
 540 FOR K=1 TO S1
 550   LET R2=R2+R3
 560   IF INT(K/12)=K/12 THEN  GOSUB 590
 570 NEXT K
 580 GOTO 600
 590 LET R3=R3*(1+A2/100): RETURN
 600 REM-CALC INVESTMENT INTEREST(I5),(I9)
 610 LET I6=I6/400: LET D2=D1: IF P3>R1 THEN  LET D3=P3-R3
 620 LET I5=0: LET I9=0
 630 FOR J=1 TO S1
 640   IF J/4<>INT(J/4) THEN 720
 650   LET I7=I6*D2
 660   LET I5=I5+I7
 670   LET D2=D2+I7
 680   IF A$<>"Y" THEN 720
 690   LET I8=I6*D4
 700   LET I9=I9+I8
 710   LET D4=D4+I8+D3
 720   IF J/12<>INT(J/12) THEN 770
 730   REM-LESS TAX
 740   LET T2=I5*T1/100: LET T3=I9*T1/100
 750   LET I5=I5-T2: LET I9=I9-T3
 760   LET D2=D2-T2: LET D4=D4-T3
 770 NEXT J
 780 REM-CALC.RENT NET COST(R4);NET MONTHLY COST(R5)
 790 IF A$<>"Y" THEN  LET I9=0
 800 LET R4=R2-I5-I9: LET R5=R4/S1
 810 REM-CALC FMV OF HOUSE
 820 LET V1=P1: LET K=S1
 830 FOR J=1 TO K
 840   LET V1=V1*(1+A1/1200)
 850 NEXT J
 860 LET V2=V1-(V1*B1/100)
 870 LET V2=V2-B2
 880 LET V3=D1+C2-V2
 890 LET V4=V3/S1
 900 REM-INFO
 910 LET P3=P3+T5
 920 PRINT TAB(10),"***BUYING INFO***"
```

```
 930 PRINT "PRICE OF HOUSE";TAB(40),%$C11F2;P1
 940 PRINT "'FRONT-MONEY'";TAB(40),%$C11F2;D1
 950 PRINT "CLOSING COSTS";TAB(40),%$C11F2;C1
 960 PRINT "MONTHLY PROP. TAXES AND INSURANCE";TAB(40),%$C11F2;T5
 970 PRINT "AMOUNT OF MORTGAGE";TAB(40),%$C11F2;P2
 980 PRINT "INTEREST RATE ON MORTGAGE";TAB(40);%C11F2;I1;" %"
 990 PRINT "MONTHLY PAYMENT (INCLUDES TAXES && INS.)";TAB(40),%$C11F2;P3
1000 INPUT "ENTER 'A' FOR COST ANALYSIS.",A$: IF A$<>"A" THEN 1000
1010 PRINT TAB(10),"**COST ANALYSIS**"
1020 PRINT "VALUE OF HOUSE AT END OF STAY";TAB(40),%$C11F2;V1
1030 PRINT "MORTGAGE BAL. AT END OF STAY";TAB(41);"(";%$C11F2;B2;%D;")"
1040 PRINT "AGENT'S COMMISSION";TAB(41);"(";%$C11F2;B1*V1/100;%D;")"
1050 PRINT "NET PROCEEDS FROM SALE";TAB(40),%$C11F2;V2
1060 PRINT TAB(10),"*************"
1070 PRINT "LENGTH OF STAY";TAB(40),S1;" MONTHS"
1080 PRINT "TOTAL OF PAYMENTS MADE";TAB(40),%$C11F2;P4+D1
1090 PRINT "TAX SAVINGS FROM BUYING";TAB(41);"(";%$C11F2;P6;")"
1100 PRINT "NET PROCEEDS FROM SALE";TAB(41);"(";%$C11F2;V2;")"
1110 PRINT "TOTAL COST OF BUYING";TAB(40),%$C11F2;V3
1120 PRINT "MONTHLY COST OF BUYING";TAB(40),%$C11F2;V4
1130 INPUT "ENTER 'R' TO DISPLAY RENTAL INFO",R$
1140 IF R$<>"R" THEN 1130
1150 PRINT TAB(10),"***RENTAL INFO***"
1160 PRINT "INITIAL MONTHLY RENT";TAB(40),%$C11F2;R1
1170 PRINT "FINAL MONTHLY RENT";TAB(40),%$C11F2;R3
1180 PRINT "TOTAL OF RENTAL PAYMENTS";TAB(40),%$C11F2;R2
1190 PRINT "INTEREST ON 'FRONT-MONEY' SAVED";TAB(40),%$C11F2;I5
1200 PRINT "INTEREST ON MTG-RENT DIFFERENCE INVESTED";TAB(40),%$C11F2;I9
1210 PRINT "NET COST OF RENTING";TAB(40),%$C11F2;R4
1220 PRINT "MONTHLY COST OF RENTING";TAB(40),%$C11F2;R5
1225 INPUT "REVIEW INFO ? (Y OR N)",Y$
1227 IF Y$="Y" THEN 920
1230 END
```

Adaptability Notes

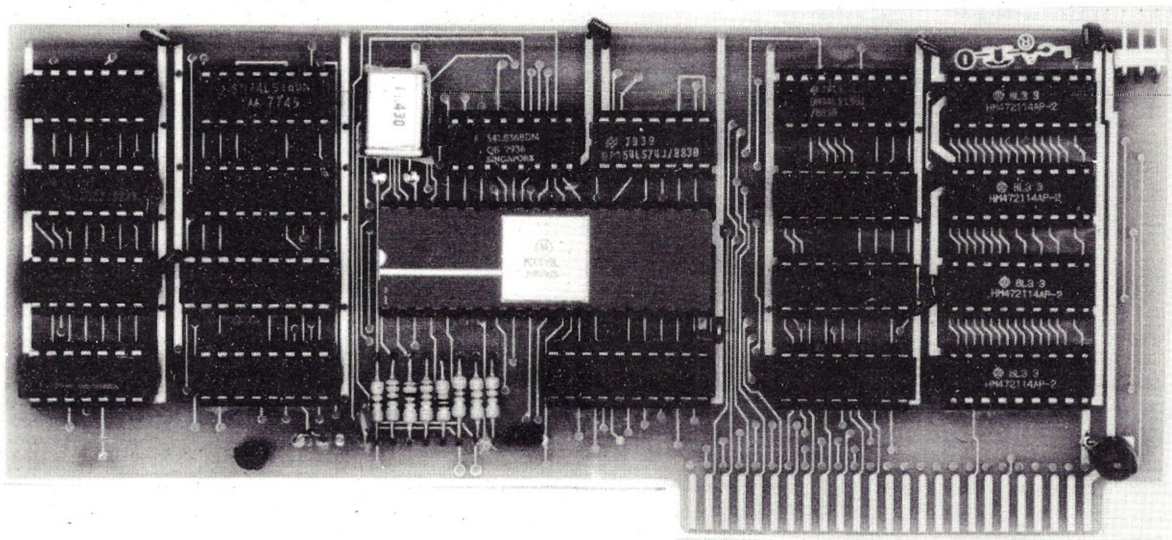
The program is written in Processor Technology Extended Cassette BASIC but is easily adapted to any BASIC.

The &Ds in line 330 display control D, which with the SOL 20 is represented by a lightning bolt. The purpose of this line is to alert the non-computer-oriented user to the fact that the momentary pause is not due to his error (and so he won't bang on the side of your CRT to get the display going again!).

In line 930 and following, the %$C11F2s are Processor Tech's formatting instructions, which cause the dollar signs, commas and proper number of decimal places to be inserted. ■

apple® computer apple® computer

DOUBLEVISION™



80 x 24 Video Display
COLUMNS LINES

with Upper and Lower Case

- Works with Apple II*, Apple II Plus*, and PASCAL
- Full 96 ASCII character set
- Fully programmable cursor:
  1-9 lines any position
  Blinking (2 speeds) and non-blinking
- All software included for BASIC (optional for PASCAL)
- Shift Lock Feature
- Built in Light Pen capability
- Inverse video
- Full cursor control
- 50/60 Hz operation
- No conflict with other boards
- Compatible with the latest in word processing software "Apple-Pie 2.0"

Available now for........... $295°°

►PASCAL software interface available for $25.00 additional
Allow up to 4 weeks for shipment.
All Mail orders add $3.00 for postage, insurance and handling

Calif. Residents add 6% Sales Tax

*Apple is a Registered TM of Apple Computers, Inc.

✓286 The Computer Stop
16919 Hawthorne Blvd.
Lawndale, CA 90260
(213) 371-4010

MON. - SAT.
10-6

J. H. Thurm
PO Box 1146
Waterdown, Ontario
Canada L0R 2H0

A COSMAC CDP1802, CDP1854 Monitor

Communicate with the CDP1802 through a UART or an ACIA with this monitor.

The monitor presented here has been written for a specific purpose. As such, it is only part of a larger program. However, I feel that it may be of sufficient interest for those who would like to communicate with their CDP1802 the way it should be done, through a UART or an ACIA.

The CDP1854 is a particularly well-suited part for this purpose, since it has two modes, one of which (Mode 1) allows it to work together with the CDP1802 with a minimum of hardware and software. It is this mode in which it is employed.

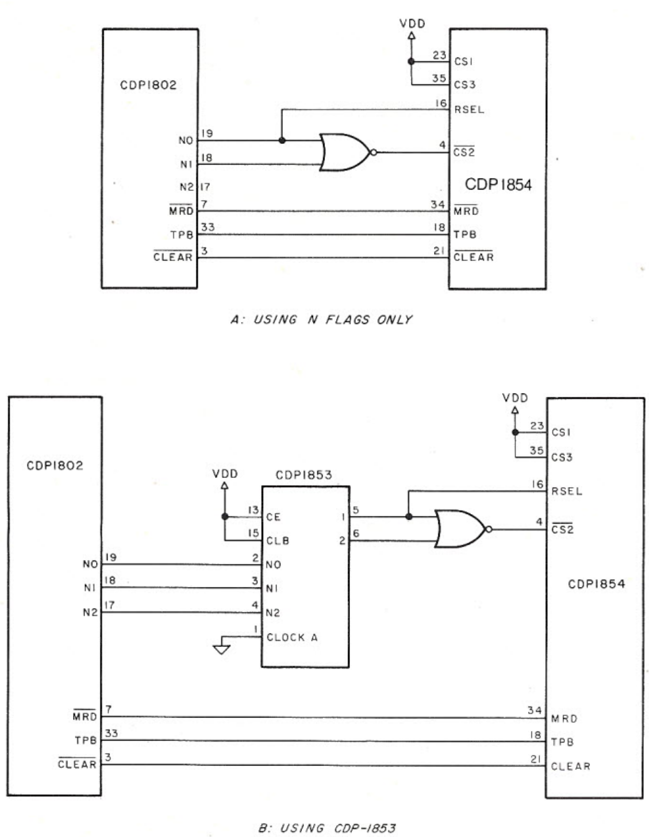


Fig. 1. Principle of addressing CDP1854.

Hardware

The monitor, in its present form, occupies 565 ROM bytes. No attempt has been made to fit the program into 512 bytes only. One of the major advantages of the COSMAC system is that it requires a single voltage only, thus you would use a 2716/2516 ROM, or at least a 2758, with plenty of spare PROM locations for your own enhancements. If a TTL-level serial terminal is used, the need for any additional voltages will be eliminated.

The monitor occupies low memory from 0000 up; RAM is required from FFFF down. With 256 RAM bytes, the system would be operational. For a minimum system, use of the highest address bit will suffice to discriminate between ROM and RAM.

The CPU communicates with the CDP1854 using the N-lines N0 and N1 and input/output commands 61, 62, 69 and 6A. Thus, the CDP1854 may be interfaced using only one additional two-input NOR gate or, if full N-line decoding is desired, a CDP-1853. For both conditions, a diagram is given (see Fig. 1). A further diagram (Fig. 2) shows connections of all relevant lines to the CDP1854.

The CDP1854 is initialized to seven data bits, no parity and two stop bits, which may alternatively be regarded as seven data bits, parity = "Mark" and one stop bit (transmitting). For other configurations, consult Fig. 9 (Control Register Bit Assignment) of the RCA CDP1854D/CD data sheet. The UART control word is byte 0028 in the monitor.

Firmware

The overriding requisite for this monitor has been the need for full, unlimited subroutine nesting ability. To accomplish this, I have used the SCRT programs found on pages 62 to 64 of the RCA User Manual for the CDP1802 COSMAC Microprocessor, Publication No. MPM-

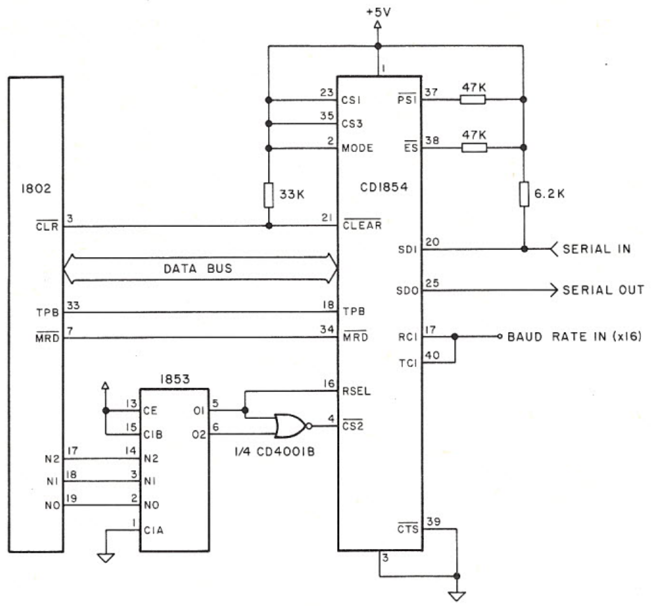


Fig. 2. Termination of all relevant pins of the CDP1854.

G	Go to an address and execute from there form: Gaaaa	
D	Dump a range of memory form: Dssss,eeee	
E	Enter hex bytes into successive memory locations from a given address form: E ssss (the space is automatically given)	
F	Fill a range of memory with a given byte form: F ssss,eeee,bb	
A	Enter a string of ASCII characters, including carriage return, line feed, bell and whatever is desired. form: A ssss Note: Entering ASCII characters must be terminated by ESC, whereupon the monitor will print out the address location of the last character entered.	
aaaa	an address	
ssss	a starting address	
eeee	an end address (inclusive)	
bb	a hex byte	

In the "Enter" command, the input is terminated by entering any non-hexadecimal character.

In entering addresses, only the first four hex characters are seen by the monitor; the next character must be a comma.

Table 1. EC-Bug commands.

201B.

I made one small addition to the SCRT programs, which should also be made by every user routine engaging the X-register, namely, resetting X to point to RF and pointing RF to FFFF before exiting. In this monitor, X *always* points to RF, and RF *always* points to FFFF, restoring these conditions whenever they have been altered by the program.

This simplifies things considerably, inasmuch as you never have to ask, "Where does X point to now?" However, you may eventually come to your own conclusions concerning this matter.

R2 is the stack pointer and is initialized to point to FFEF. The top 16 locations of RAM are reserved for input-output and computed message strings (not presently employed by the monitor). Registers 1 and A are not yet utilized but are initialized to 0000, R1 by the program, RA by default, and may be employed by the user, as well as RB, RC and, for the most part, RD.

RE.1 is used as a shuttle to hand characters from one subroutine to another and back.

The monitor contains a string-output routine (STOUT) to facilitate the transmission of ASCII strings of any length. Set RF to the head of string, make sure RX points to RF, then call STOUT. STOUT will reset RF before exiting. The *last* character of every message must have a 1 in bit 7 to tell STOUT it is finished. STOUT may also be used to transmit single characters, such as in ALF, ASP and APR.

Please note that the 16 bytes of ATABL *must* be at the *head of a page* for the routines HEXIT and ASCIT to function.

The assembly has been done on a somewhat primitive cross assembler. Please note that the asterisk signifies a high-order address byte. Calling a subroutine is therefore done as:

```
SEP   R4
DB*   LABEL
DB    LABEL
```

Commands

In its present state, the monitor will allow four commands but is written so that more may easily be added. These commands are:

"D" = Dump, or display

"E" = Enter HEX Bytes

```
EC-BUG HEX LIST:

> D 0000,0220

0000: F8 00 B1 F8 00 A1 F8 00 B4 F8 71 A4 F8 00 B5 F8
0010: 82 A5 F8 00 B7 F8 6A A7 F8 00 B6 A6 F8 FF BF AF
0020: F8 FF B2 F8 EF A2 EF F8 15 5F 61 2F F8 00 B3 F8
0030: 33 A3 D3 D4 00 BF D4 00 A7 D4 00 9B D4 00 8F 9E
0040: FB 41 C2 01 EF FB 05 C2 01 78 FB 01 C2 01 9F FB
0050: 03 C2 01 C4 FB 01 C2 01 DD FB 1F 32 64 FB 02 C2
0060: 08 00 30 3C D4 00 AB 30 18 D3 69 FE 3B 6A 30 69
0070: D3 E2 96 73 86 73 93 B6 83 A6 46 B3 46 A3 EF 30
0080: 70 D3 96 B3 86 A3 E2 12 72 A6 F0 B6 EF 30 81 69
0090: F6 3B 8F 6A BE D7 9E 5F 62 2F D5 F8 20 30 AD F8
00A0: 0A 30 AD F8 3A 30 AD F8 3E 30 AD F8 2A BE D4 00
00B0: 95 D5 0D BE D4 00 95 0D FE 33 BE 1D 30 B2 D5 F8
00C0: 00 BD F8 C7 AD 30 B2 0D 0A 7F 7F 7F FF D4 00 8F
00D0: D4 00 E5 9E FE FE FE FE AE D4 00 8F D4 00 E5 9E
00E0: 5F 8E F1 BE D5 F8 01 BF F8 00 AF 9E F3 32 F7 8F
00F0: FB 0F 32 18 1F 30 EB 8F BE F8 FF BF AF D5 C4 C4
0100: 30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46
0110: F8 01 BF 9E AF 0F BE F8 FF BF AF D5 9E AE F6 F6
0120: F6 F6 BE D4 01 10 D4 00 95 8E FA 0F BE D4 01 10
0130: D4 00 95 D5 D4 00 CD 9E BD D4 00 CD 9E AD D5 D4
0140: 01 34 9D B8 8D A8 D4 00 8F 9E FB 2C 3A 46 D4 01
0150: 34 9D B9 8D A9 D5 EF 99 5F 98 F3 3A 66 89 5F 88
0160: F3 3A 66 C0 00 18 D5 98 BE D4 01 1C 88 BE D4 01
0170: 1C D4 00 A3 D4 00 9B D5 D4 00 9B D4 01 3F D4 00
0180: 9F D4 00 BF D4 01 67 08 BE D4 01 1C D4 01 56 88
0190: FA 0F FB 0F 32 9C D4 00 9B 18 30 87 18 30 81 D4
01A0: 00 9B D4 01 34 9D B8 8D A8 D4 00 BF D4 01 67 D4
01B0: 00 CD 9E 58 D4 00 9B 88 FA 0F FB 0F 32 C1 18 30
01C0: AF 18 30 A9 D4 00 9B D4 01 3F D4 00 8F 9E FB 2C
01D0: 3A CA D4 00 CD 9E 58 D4 01 56 18 30 D5 D4 00 9B
01E0: D4 01 34 F8 01 B9 F8 EA A9 D9 9D B3 8D A3 D3 D4
01F0: 00 9B D4 01 34 9D B8 8D A8 D4 00 BF D4 00 9F D4
0200: 00 8F 9E FB 1B 32 0D 9E 58 18 C0 01 FF 28 08 F9
0210: 80 58 D4 00 8F D4 00 A3 D4 01 67 D4 00 9F C0 00
0220: 64
>
```

Hex listing of EC-Bug monitor (done on a TI terminal).

EC-Bug monitor listing.

```
G100
COSMAC 1802 ASSEMBLER
SOURCE FILE? ECBUG,2000

SYMBOL TABLE

CWORD 0015
STAT  0001
CNTRL 0001
STACK FFEF
DATA  0002
RAM   0800
INTRR 0000
BEGIN 0000
```

Label	Address	
INCH	0059	
OUTCH	005F	
ALF	0065	
ASP	005D	
APR	0075	
STOUT	0080	
CRLF	008C	
BIN	00C5	
HEXIT	00DD	
ASCIT	012C	
BOUT	0138	
OUTA	0150	
COMPA	0164	(will exit on match)
DADD	01AC	
ADIN	01E7	

Addresses to Use as Program Exits

CSTART	0000
WSTRT	001E
REPR	009F

Table 2. Subroutine that may be called by the user (see assembly listing for details).

```
DEMONSTRATION OF EC-BUG COMMANDS:

> F 0000,0FFF,00
> E 0800
0800: F8 09 BD F8 00 AD D4 00 B2 D5 .
> A 0900

     THE QUICK BROWN FOX JUMPS
     OVER THE LAZY DOG.
:0940:
         ◆
> D 0900,0943

0900: 0D 0A 7F 7F 7F 7F 20 20 20 20 20 54 48 45 20 51
0910: 55 49 43 4B 20 42 52 4F 57 4E 20 46 4F 58 20 4A
0920: 55 4D 50 53 0D 0A 7F 7F 7F 7F 20 20 20 20 20 4F
0930: 56 45 52 20 54 48 45 20 4C 41 5A 59 20 44 4F 47
0940: AE 00 00 00
> G 0800
     THE QUICK BROWN FOX JUMPS
     OVER THE LAZY DOG.
> Z
     THE QUICK BROWN FOX JUMPS
     OVER THE LAZY DOG.
>

NOTE: THE "Z" COMMAND CAUSES THE MONITOR TO GO
      DIRECTLY TO MEMORY LOCATION 0800 WITHOUT
      THE NEED TO ENTER AN ADDRESS.
```

Sample run. EC-Bug commands.

"F" = Fill Memory
"G" = Go and execute
Also See Table 1.

Conclusion

For some applications (for instance, remote monitoring of soil moisture with solar cells as the only power source), C/MOS is the only choice. In addition, the C/MOS structure can inherently tolerate far greater temperature extremes than other processes.

I hope that this brief presentation will win new friends for the COSMAC CDP1802, showing that it can do everything other processors can do and, almost as easily, some things even better. I also hope that this article may alleviate in potential users some of the apprehensions generated by the peculiar instruction set of the COSMAC.

Once you can do unlimited subroutine nesting, you've got it made (or almost). ■

```
REST   0018
INIT   0027
PRMPT  0033
CMDLP  003C
STAR   0064
GO     0069
LOOK   006A
EXITA  0070
CALL   0071
EXITR  0081
RETRN  0082
INCH   008F
OUTCH  0095
ASP    009B
ALF    009F
ACL    00A3
APR    00A7
AST    00AB
ONEC   00AD
STOUT  00B2
FINIS  00BE
CRLF   00BF
CLN    00C7
BIN    00CD
HEXIT  00E5
MHEX   00EB
GOT    00F7
ATABL  0100
ASCIT  0110
BOUT   011C
ADIN   0134
DADD   013F
NOC    0146
COMPA  0156
NYET   0166
OUTA   0167
DUMP   0178
STLN   0181
ANUB   0187
PFNL   019C
ENTER  019F
NULIN  01A9
NUBYT  01AF
PRNL   01C1
FILL   01C4
GCOM   01CA
FMOR   01D5
GOTO   01DD
SWPR   01EA
STENT  01EF
CHIN   01FF
BREK   020D

                ;ECBUG  A MONITOR FOR THE COSMAC CDP-1802
                ;       IN CONJUNCTION WITH THE CDP-1854 UART.
                ;       VER. 2.0         JANUARY 1979.
                ;       BY J.H. THURM.

0000                    ORG    0
0000                    CWORD  EQU    15      ;UART INITIALIZE
0000                    STAT   EQU    1       ;UART    STATUS
0000                    CNTRL  EQU    1       ;UART CONTROL
0000                    STACK  EQU    0FFEF   ;STACK POINTER
0000                    DATA   EQU    2       ;DATA    I/O
0000                    RAM    EQU    0800
0000                    INTRR  EQU    0       ;INTERRUPT DUMMY

0000 F800       BEGIN   LDI*   INTRR   ;INITIALIZE REGISTERS
0002 B1                 PHI    R1
0003 F800               LDI    INTRR
0005 A1                 PLO    R1
0006 F800               LDI*   CALL
0008 B4                 PHI    R4
0009 F871               LDI    CALL
000B A4                 PLO    R4
000C F800               LDI*   RETRN
000E B5                 PHI    R5
```

```
000F F882          LDI    RETRN
0011 A5            PLO    R5
0012 F800          LDI*   LOOK
0014 B7            PHI    R7
0015 F86A          LDI    LOOK
0017 A7            PLO    R7
0018 F800   REST   LDI    00       ;DE-RANDOMIZE   R6
001A B6            PHI    R6
001B A6            PLO    R6
001C F8FF          LDI    0FF
001E BF            PHI    RF
001F AF            PLO    RF
0020 F8FF          LDI*   STACK
0022 B2            PHI    R2
0023 F8EF          LDI    STACK
0025 A2            PLO    R2
0026 EF            SEX    RF

0027 F815   INIT   LDI    CWORD    ;INITIALIZE UART
0029 5F            STR    RF
002A 61            OUT    CNTRL
002B 2F            DEC    RF       ;RESTORE RF

002C F800          LDI*   PRMPT
002E B3            PHI    R3
002F F833          LDI    PRMPT
0031 A3            PLO    R3
0032 D3            SEP    R3

0033 D4     PRMPT  SEP    R4       ;DO A CR/LF
0034 00            DB*    CRLF
0035 BF            DB     CRLF
0036 D4            SEP    R4       ;PRINT  PROMPT SIGN
0037 00            DB*    APR
0038 A7            DB     APR
0039 D4            SEP    R4       ;DO A SPACE
003A 00            DB*    ASP
003B 9B            DB     ASP

003C D4     CMDLP  SEP    R4       ;GET    COMMAND
003D 00            DB*    INCH
003E 8F            DB     INCH
003F 9E            GHI    RE       ;GET    CHARACTER
0040 FB41          XRI    41       ;A?
0042 C201EF        LBZ    STENT
0045 FB05          XRI    05       ;D?
0047 C20178        LBZ    DUMP
004A FB01          XRI    01       ;E?
004C C2019F        LBZ    ENTER
004F FB03          XRI    03       ;F?
0051 C201C4        LBZ    FILL
0054 FB01          XRI    01       ;G?
0056 C201DD        LBZ    GOTO
0059 FB1F          XRI    1F       ;X?
005B 3264          BZ     STAR
005D FB02          XRI    02       ;Z?
005F C20800        LBZ    RAM
0062 303C          BR     CMDLP

0064 D4     STAR   SEP    R4       ;PRINT  STAR TO INDICATE
0065 00            DB*    AST      ;BREAKOUT (ILLEGAL ENTRY)
0066 AB            DB     AST
0067 3018          BR     REST

            ;LOOK  SCANS THRE OF UART

0069 D3     GO     SEP    R3
006A 69     LOOK   INP    STAT
006B FE            SHL             ;EMPTY?
006C 3B6A          BNF    LOOK     ;LOOP, IF NOT EMPTY
006E 3069          BR     GO

            ;STANDART CALL AND RETURN ROUTINES

0070 D3     EXITA  SEP    R3       ;RETURN
0071 E2     CALL   SEX    R2
0072 96            GHI    R6
0073 73            STXD
0074 86            GLO    R6
0075 73            STXD
0076 93            GHI    R3
0077 B6            PHI    R6
0078 83            GLO    R3
0079 A6            PLO    R6
007A 46            LDA    R6
007B B3            PHI    R3
007C 46            LDA    R6
007D A3            PLO    R3
007E EF            SEX    RF
007F 3070          BR     EXITA

0081 D3     EXITR  SEP    R3
0082 96     RETRN  GHI    R6
0083 B3            PHI    R3
0084 86            GLO    R6
0085 A3            PLO    R3
0086 E2            SEX    R2
0087 12            INC    R2
0088 72            LDXA
0089 A6            PLO    R6
008A F0            LDX
008B B6            PHI    R6
008C EF            SEX    RF
008D 3081          BR     EXITR

008F 69     INCH   INP    STAT     ;WAIT FOR TERMINAL ENTRY
0090 F6            SHR
0091 3B8F          BNF    INCH     ;LOOP UNTIL ENTRY
0093 6A            INP    DATA     ;CHAR. ARRIVED
0094 BE            PHI    RE       ;STORE IT FOR CALLING ROUTINE

0095 D7     OUTCH  SEP    R7       ;CALL LOOK
0096 9E            GHI    RE       ;GET CHARACTER
0097 5F            STR    RF
0098 62            OUT    DATA     ;ECHO IF IT WAS INPUT
0099 2F            DEC    RF       ;RESTORE RF
009A D5            SEP    R5       ;RETURN

009B F820   ASP    LDI    20       ;PRINTS A SPACE
009D 30AD          BR     ONEC
009F F80A   ALF    LDI    0A       ;PRINTS A LINE FEED
00A1 30AD          BR     ONEC
00A3 F83A   ACL    LDI    3A       ;PRINTS A COLON
00A5 30AD          BR     ONEC
00A7 F83E   APR    LDI    3E       ;PRINTS A PROMPT SIGN
00A9 30AD          BR     ONEC
00AB F82A   AST    LDI    2A       ;PRINTS A STAR
00AD BE     ONEC   PHI    RE
00AE D4            SEP    R4
00AF 00            DB*    OUTCH
00B0 95            DB     OUTCH
00B1 D5            SEP    R5       ;RETURN

            ;STOUT   STRING OUTPUT ROUTINE
            ;EXPECTS RD TO POINT TO TOP OF
            ;STRING. LAST CHARACTER IN STRING
            ;MUST HAVE BIT 7=1.

00B2 0D     STOUT  LDN    RD       ;GET CHARACTER
00B3 BE            PHI    RE       ;PUT INTO SHUTTLE
00B4 D4            SEP    R4       ;GO TO PRINT IT
00B5 00            DB*    OUTCH
00B6 95            DB     OUTCH
00B7 0D            LDN    RD       ;GET CHARACTER AGAIN
00B8 FE            SHL             ;BIT 7=1?
00B9 33BE          BDF    FINIS    ;IF YES, RETURN
00BB 1D            INC    RD       ;ADVANCE POINTER
00BC 30B2          BR     STOUT    ;DO NEXT CHARACTER
00BE D5     FINIS  SEP    R5       ;RETURN

            ;CRLF  DOES CARRIAGE RETURN, LINE FEED
            ;AND NULLS FOR DELAY

00BF F800   CRLF   LDI*   CLN      ;SET RD TO
00C1 BD            PHI    RD       ;HEAD OF TABLE
```

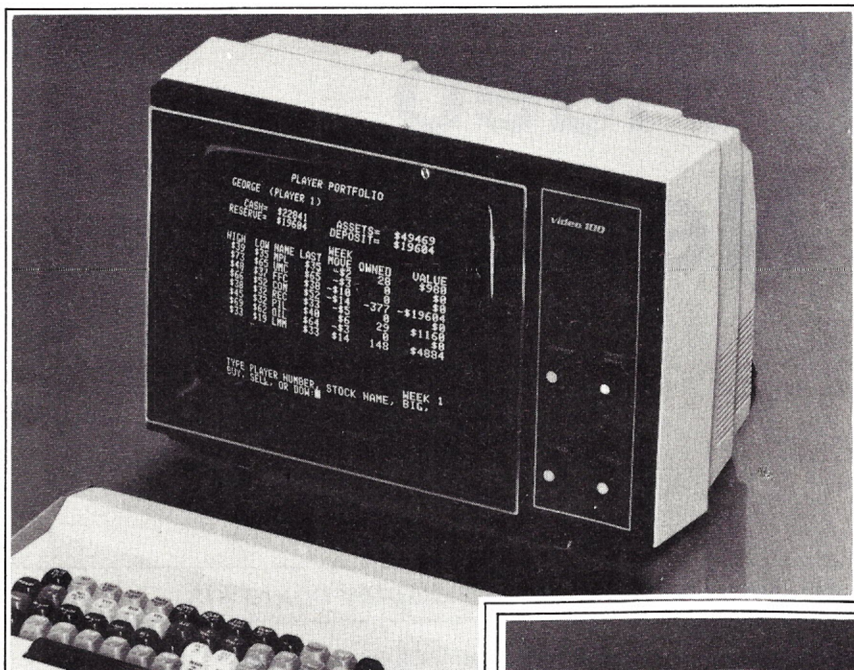
```
00C2 F8C7           LDI     CLN
00C4 AD             PLO     RD
00C5 30B2           BR      STOUT

00C7 0D     CLN     DB      0D      ;CR
00C8 0A             DB      0A      ;LF
00C9 7F             DB      7F      ;DEL
00CA 7F             DB      7F      ;DEL
00CB 7F             DB      7F      ;DEL
00CC FF             DB      0FF     ;DEL, END OF STRING


            ;BIN, A SUBROUTINE TO ENTER TWO HEX-CHARACTERS
            ;AND ASSEMBLE THEM INTO A BYTE

00CD D4     BIN     SEP     R4      ;GET FIRST CHARACTER
00CE 00             DB*     INCH
00CF 8F             DB      INCH
00D0 D4             SEP     R4      ;GO TO CONVERT IT
00D1 00             DB*     HEXIT
00D2 E5             DB      HEXIT
00D3 9E             GHI     RE      ;GET HEX NIBL
00D4 FE             SHL             ;SHIFT IT TO HIGH NIBL
00D5 FE             SHL
00D6 FE             SHL
00D7 FE             SHL
00D8 AE             PLO     RE      ;SAVE IT FOR NOW
00D9 D4             SEP     R4      ;GET SECOND CHARACTER
00DA 00             DB*     INCH
00DB 8F             DB      INCH
00DC D4             SEP     R4      ;CONVERT IT
00DD 00             DB*     HEXIT
00DE E5             DB      HEXIT
00DF 9E             GHI     RE      ;GET LOW NIBL
00E0 5F             STR     RF      ;STORE IT FOR ORING
00E1 8E             GLO     RE      ;RECALL HIGH NIBL
00E2 F1             OR              ;OR IT TO LOW NIBL
00E3 BE             PHI     RE      ;LEAVI IT FOR CALLING ROUTINE
00E4 D5             SEP     R5      ;RETURN

00E5 F801   HEXIT   LDI*    ATABL   ;SET RF TO POINT TO TABLE
00E7 BF             PHI     RF
00E8 F800           LDI     ATABL
00EA AF             PLO     RF
00EB 9E     MHEX    GHI     RE      ;GET CHARACTER
00EC F3             XOR             ;COMPARE
00ED 32F7           BZ      GOT     ;EXIT, IF MATCH FOUND
00EF 8F             GLO     RF      ;GET LOW ADDRESS
00F0 FB0F           XRI     0F      ;COMPARE FOR END OF TABLE
00F2 3218           BZ      REST    ;OUT, IF NO MATCH FOUND
00F4 1F             INC     RF      ;ADVANCE POINTER
00F5 30EB           BR      MHEX    ;DO NEXT COMPARISON
00F7 8F     GOT     GLO     RF      ;GET LOW ADDRESS
00F8 BE             PHI     RE      ;PUT IT INTO SHUTTLE
00F9 F8FF           LDI     0FF     ;RESTORE RF
00FB BF             PHI     RF
00FC AF             PLO     RF
00FD D5             SEP     R5      ;RETURN

00FE C4             NOP
00FF C4             NOP

0100 30     ATABL   DB      30
0101 31             DB      31
0102 32             DB      32
0103 33             DB      33
0104 34             DB      34
0105 35             DB      35
0106 36             DB      36
0107 37             DB      37
0108 38             DB      38
0109 39             DB      39
010A 41             DB      41
010B 42             DB      42
010C 43             DB      43
010D 44             DB      44
010E 45             DB      45
010F 46             DB      46

            ;ASCIT   SUBROUTINE TO RECEIVE A HEX NIBL
            ;        IN RE.1, CONVERT IT INTO AN ASCII-CHARACTER,
            ;        AND RETURN IT TO CALLING ROUTINE IN RE.1

0110 F801   ASCIT   LDI*    ATABL   ;SET RF TO POINT
0112 BF             PHI     RF      ;TO ASCII CHARACTER
0113 9E             GHI     RE
0114 AF             PLO     RF
0115 0F             LDN     RF      ;GET CHARACTER OPPOSITE ADDRESS
0116 BE             PHI     RE
0117 F8FF           LDI     0FF     ;RESTORE RF
0119 BF             PHI     RF
011A AF             PLO     RF
011B D5             SEP     R5      ;RETURN


            ;BOUT    SUBROUTINE TO PRINT A BYTE
            ;        RECEIVED IN RE.1

011C 9E     BOUT    GHI     RE      ;GET BYTE
011D AE             PLO     RE      ;SAVE IT FOR LOWER NIBL
011E F6             SHR
011F F6             SHR
0120 F6             SHR
0121 F6             SHR
0122 BE             PHI     RE
0123 D4             SEP     R4      ;CONVERT IT
0124 01             DB*     ASCIT
0125 10             DB      ASCIT
0126 D4             SEP     R4      ;ASCII IS IN RE.1!
0127 00             DB*     OUTCH
0128 95             DB      OUTCH
0129 8E             GLO     RE      ;PROCESS NOW LOWER NIBL
012A FA0F           ANI     0F      ;MASK OUT UPPER NIBL
012C BE             PHI     RE
012D D4             SEP     R4
012E 01             DB*     ASCIT
012F 10             DB      ASCIT
0130 D4             SEP     R4
0131 00             DB*     OUTCH
0132 95             DB      OUTCH
0133 D5             SEP     R5

            ;ADIN    ASSEMBLES AN ADDRESS AND LEAVES IT IN RD

0134 D4     ADIN    SEP     R4      ;GET HIGH BYTE OF ADDRESS
0135 00             DB*     BIN
0136 CD             DB      BIN
0137 9E             GHI     RE      ;BYTE IS IN RE.1
0138 BD             PHI     RD      ;MAKE IT HIGH ADDRESS BYTE
0139 D4             SEP     R4      ;GET LOW BYTE OF ADDRESS
013A 00             DB*     BIN
013B CD             DB      BIN
013C 9E             GHI     RE
013D AD             PLO     RD      ;MAKE IT LOW BYTE OF ADDRESS
013E D5             SEP     R5      ;RETURN


            ;DADD    GETS A DOUBLE ADDRESS, LEAVES HIGH ADDRESS IN R8
            ;        AND LEAVES LOW ADDRESS IN R9

013F D4     DADD    SEP     R4      ;GET FIRST ADDRESS
0140 01             DB*     ADIN
0141 34             DB      ADIN
0142 9D             GHI     RD      ;PUT INTO R8
0143 B8             PHI     R8
0144 8D             GLO     RD
0145 A8             PLO     R8
0146 D4     NOC     SEP     R4      ;GET A COMMA
0147 00             DB*     INCH
0148 8F             DB      INCH
0149 9E             GHI     RE      ;WAS IT A COMMA?
014A FB2C           XRI     2C
014C 3A46           BNZ     NOC     ;IF NOT TRY AGAIN
014E D4             SEP     R4      ;GET SECOND ADDRESS
014F 01             DB*     ADIN
```

✓Reader Service—see page 258

```
0150 34                 DB      ADIN
0151 9D                 GHI     RD      ;PUT IT INTO R9
0152 B9                 PHI     R9
0153 8D                 GLO     RD
0154 A9                 PLO     R9
0155 D5                 SEP     R5      ;RETURN


            ;COMPA      COMPARES THE ADDRESS IN R8 TO THE
            ;           ADDRESS IN R9. JUMPS TO REST
            ;           IF MATCH IS FOUND.

0156 EF     COMPA       SEX     RF
0157 99                 GHI     R9
0158 5F                 STR     RF
0159 98                 GHI     R8
015A F3                 XOR             ;COMPARE HIGH BYTES
015B 3A66               BNZ     NYET
015D 89                 GLO     R9
015E 5F                 STR     RF
015F 88                 GLO     R8
0160 F3                 XOR             ;COMPARE LOW BYTES
0161 3A66               BNZ     NYET
0163 C00018             LBR     REST
0166 D5     NYET        SEP     R5      ;RETURN


            ;OUTA       PRINTS THE ADDRESS IN R8

0167 98     OUTA        GHI     R8      ;GET HIGH BYTE OF ADDRESS
0168 BE                 PHI     RE      ;PUT IT INTO SHUTTLE
0169 D4                 SEP     R4      ;PRINT IT
016A 01                 DB*     BOUT
016B 1C                 DB      BOUT
016C 88                 GLO     R8      ;GET LOW BYTE OF ADDRESS
016D BE                 PHI     RE      ;E.T.C.
016E D4                 SEP     R4
016F 01                 DB*     BOUT
0170 1C                 DB      BOUT
0171 D4                 SEP     R4      ;PRINT A COLON
0172 00                 DB*     ACL
0173 A3                 DB      ACL
0174 D4                 SEP     R4      ;PRINT A SPACE
0175 00                 DB*     ASP
0176 9B                 DB      ASP
0177 D5                 SEP     R5      ;RETURN


            ;DUMP       DUMPS BYTES FROM A STARTING ADDRESS
            ;           TO AN END ADDRESS

0178 D4     DUMP        SEP     R4      ;DO A SPACE
0179 00                 DB*     ASP
017A 9B                 DB      ASP
017B D4                 SEP     R4      ;GET ADDRESSES
017C 01                 DB*     DADD
017D 3F                 DB      DADD
017E D4                 SEP     R4      ;DO A LINE FEED
017F 00                 DB*     ALF     ;FOR CLARITY
0180 9F                 DB      ALF
0181 D4     STLN        SEP     R4      ;START THE LINE
0182 00                 DB*     CRLF
0183 BF                 DB      CRLF
0184 D4                 SEP     R4      ;PRINT THE ADDRESS
0185 01                 DB*     OUTA
0186 67                 DB      OUTA
0187 08     ANUB        LDN     R8      ;LOAD THE BYTE
0188 BE                 PHI     RE      ;PUT IT INTO SHUTTLE
0189 D4                 SEP     R4      ;PRINT IT
018A 01                 DB*     BOUT
018B 1C                 DB      BOUT
018C D4                 SEP     R4      ;COMPARE FOR LAST ADDRESS
018D 01                 DB*     COMPA
018E 56                 DB      COMPA
018F 88                 GLO     R8      ;COMPARE FOR END OF LINE
0190 FA0F               ANI     0F
0192 FB0F               XRI     0F
0194 329C               BZ      PFNL    ;IF NOT END,NEXT BYTE OUT
0196 D4                 SEP     R4      ;PRINT A SPACE
0197 00                 DB*     ASP
0198 9B                 DB      ASP
0199 18                 INC     R8      ;ADVANCE POINTER
019A 3087               BR      ANUB
019C 18     PFNL        INC     R8
019D 3081               BR      STLN


            ;ENTER      LOADS HEX BYTES INTO SUCCESSIVE MEMORY LOCATIONS
            ;           FROM A STARTING ADDRESS. RETURNS UPON ENTRY
            ;           OF ANY ILLEGAL CHARACTER

019F D4     ENTER       SEP     R4      ;DO A SPACE FOR CLARITY
01A0 00                 DB*     ASP
01A1 9B                 DB      ASP
01A2 D4                 SEP     R4      ;GET THE ADDRESS
01A3 01                 DB*     ADIN
01A4 34                 DB      ADIN
01A5 9D                 GHI     RD
01A6 B8                 PHI     R8
01A7 8D                 GLO     RD
01A8 A8                 PLO     R8
01A9 D4     NULIN       SEP     R4      ;DO CR/LF
01AA 00                 DB*     CRLF
01AB BF                 DB      CRLF
01AC D4                 SEP     R4      ;PRINT THE ADDRESS
01AD 01                 DB*     OUTA
01AE 67                 DB      OUTA
01AF D4     NUBYT       SEP     R4      ;GET BYTE
01B0 00                 DB*     BIN
01B1 CD                 DB      BIN
01B2 9E                 GHI     RE      ;BYTE IS IN RE.1
01B3 58                 STR     R8      ;DEPOSIT
01B4 D4                 SEP     R4      ;PRINT A SPACE
01B5 00                 DB*     ASP
01B6 9B                 DB      ASP
01B7 88                 GLO     R8      ;CHECK FOR END OF LINE
01B8 FA0F               ANI     0F
01BA FB0F               XRI     0F
01BC 32C1               BZ      PRNL
01BE 18                 INC     R8      ;ADVANCE POINTER
01BF 30AF               BR      NUBYT
01C1 18     PRNL        INC     R8
01C2 30A9               BR      NULIN


            ;FILL       FILLS A DEFINED RANGE OF MEMORY
            ;           WITH THE SAME BYTE

01C4 D4     FILL        SEP     R4      ;DO A SPACE
01C5 00                 DB*     ASP
01C6 9B                 DB      ASP
01C7 D4                 SEP     R4      ;GET  ADDRESS RANGE
01C8 01                 DB*     DADD
01C9 3F                 DB      DADD
01CA D4     GCOM        SEP     R4      ;GET A COMMA
01CB 00                 DB*     INCH
01CC 8F                 DB      INCH
01CD 9E                 GHI     RE      ;GET CHARACTER
01CE FB2C               XRI     2C      ;WAS IT A COMMA
01D0 3ACA               BNZ     GCOM    ;IF NOT, TRY AGAIN
01D2 D4                 SEP     R4      ;GET FILL BYTE
01D3 00                 DB*     BIN
01D4 CD                 DB      BIN
01D5 9E     FMOR        GHI     RE
01D6 58                 STR     R8      ;STORE IT TO RAM
01D7 D4                 SEP     R4      ;COMPARE FOR ADDRESS LIMIT
01D8 01                 DB*     COMPA
01D9 56                 DB      COMPA
01DA 18                 INC     R8      ;ADVANCE POINTER
01DB 30D5               BR      FMOR
```

```
                ;GOTO   STARTS TO EXECUTE FROM ENTERED ADDRESS
01DD D4         GOTO    SEP     R4      ;DO A SPACE
01DE 00                 DB*     ASP
01DF 9B                 DB      ASP
01E0 D4                 SEP     R4      ;GET THE ADDRESS
01E1 01                 DB*     ADIN
01E2 34                 DB      ADIN
01E3 F801               LDI*    SWPR    ;PREPARE TO FREE R3
01E5 B9                 PHI     R9
01E6 F8EA               LDI     SWPR
01E8 A9                 PLO     R9
01E9 D9                 SEP     R9      ;SWAP PROGRAM COUNTERS
01EA 9D         SWPR    GHI     RD      ;ADDRESS IS IN RD!
01EB B3                 PHI     R3      ;PUT IT INTO R3
01EC 8D                 GLO     RD
01ED A3                 PLO     R3
01EE D3                 SEP     R3      ;GO TO ADDRESS

                ;STENT  A ROUTINE TO ENTER A STRING OF ASCII-CHARACTERS
                ;       INTO SUCCESSIVE MEMORY LOCATIONS.AFTER ENTERING
                ;       'ESC' IT WILL SET BIT 7 OF THE LAST CHARACTER
                ;       ENTERED TO 1 AND PRINT THE ADDRESS OF THE
                ;       THE LAST CHARACTER ENTERED.

01EF D4         STENT   SEP     R4      ;PRINT A SPACE
01F0 00                 DB*     ASP
01F1 9B                 DB      ASP
01F2 D4                 SEP     R4      ;GET THE ADDRESS
01F3 01                 DB*     ADIN
01F4 34                 DB      ADIN
01F5 9D                 GHI     RD      ;TRANSFER IT TO R8
01F6 B8                 PHI     R8
01F7 8D                 GLO     RD
01F8 A8                 PLO     R8
01F9 D4                 SEP     R4      ;DO A CR/LF
01FA 00                 DB*     CRLF
01FB BF                 DB      CRLF
01FC D4                 SEP     R4      ;ONE MORE LINE FEED
01FD 00                 DB*     ALF
01FE 9F                 DB      ALF
01FF D4         CHIN    SEP     R4      ;GET INPUT
0200 00                 DB*     INCH
0201 8F                 DB      INCH
0202 9E                 GHI     RE      ;GET    CHAR.
0203 FB1B               XRI     1B      ;WAS IT AN 'ESC;?
0205 320D               BZ      BREK    ;IF YES, EXIT
0207 9E                 GHI     RE      ;GET CHAR. AGAIN
0208 58                 STR     R8      ;DEPOSIT TO RAM
0209 18                 INC     R8      ;ADVANCE POINTER
020A C001FF             LBR     CHIN    ;NEXT INPUT
020D 28         BREK    DEC     R8      ;POINT BACK TO LAST INPUT
020E 08                 LDN     R8      ;GET IT
020F F980               ORI     80      ;SET BIT 7=1
0211 58                 STR     R8      ;PUT IT BACK
0212 D4                 SEP     R4      ;DO CR/LF
0213 00                 DB*     CRLF
0214 BF                 DB      CRLF
0215 D4                 SEP     R4      ;DO A COLON
0216 00                 DB*     ACL
0217 A3                 DB      ACL
0218 D4                 SEP     R4      ;PRINT THE ADDRESS
0219 01                 DB*     OUTA    ;OF LAST CHAR. ENTERED
021A 67                 DB      OUTA
021B D4                 SEP     R4      ;DO A LINE FEED
021C 00                 DB*     ALF     ;FOR CLARITY
021D 9F                 DB      ALF
021E C00064             LBR     STAR    ;BACK TO MONITOR

0221                    END.

VER 1.0
```

PROFESSIONAL B/W MONITORS
Designed for industry...priced for the home.

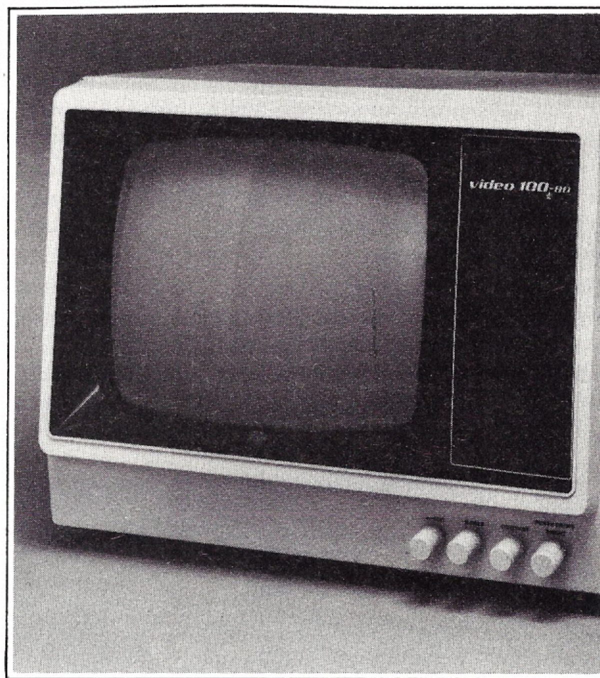


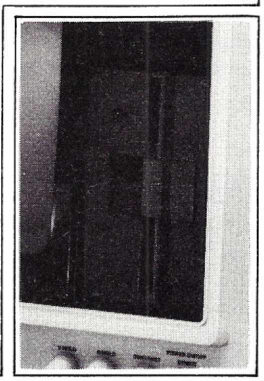
video 100

The video 100 computer monitors are ideal for all your personal and business needs. These highly reliable 12" black and white monitors feature a 12 MHz band width and 80 character by 24 line display. Plug-in compatability with Apple, Atari, Radio Shack, O.S.I., Micro-Term and Exidy make these the perfect text display for almost any system.

Sturdy, lightweight plastic cabinet

UNDER $160.00





video 100-80

The model 80 features an industrial grade metal cabinet with built-in disk mounting capability and space for an 11" x 14" PC board for custom designed electronics.

The solid state circuitry assures a sharp, stable, and trouble-free picture. The front panel controls include power, contrast, horizontal hold, vertical hold, and brightness. Adjustments for size, video level, and width are located on the rear panel.

Rugged metal cabinet with disk space

UNDER $200.00

VIDEO 100 AND VIDEO 100-80 SPECIFICATIONS

- 12" diagonal measure display
- Convenient front panel controls
- Video bandwidth 12MHz ±3 DB
- Input impedance 75 Ohms
- 80 character by 24 line display
- 90° deflection picture tube
- Video 100-80 provides mounting space for mini floppy disk.
- Resolution—Over 700 lines at center horizontally—over 350 lines at center vertically

LEEDEX CORPORATION ✓59

2420 East Oakton • Suite E • Arlington Heights, Ill. 60005 (312) 364-1180 • TLX: 25-4786

Dealer discount available

Elf Video from Netronics

Hit the big time with the Netronics R&D video display board kit.

Donald R. Shroyer
209 Brinker St.
Latrobe PA 15650

If it's time to improve your Elf computer system, consider the Netronics R&D video display board kit. It is available as a complete stand-alone terminal with ASCII keyboard or as a display board compatible with most ASCII-encoded keyboards. I have satisfactorily mated my video board with a Radio Shack project board keyboard.

The Netronics product has many features that belie its modest $89.95 (kit without keyboard) price tag:

- F-8 microprocessor controlled with crystal oscillator
- Memory and character generator, so no memory or I/O mapping is required
- Jumper-selectable ASCII or five-level signal source
- Jumper-selectable baud rates—110 or 300 for ASCII, 45.45 or 74.2 for five-level
- Jumper-selectable 50 or 60 Hz operation
- Display of 16 vertical by 32 or 64 horizontal characters, jumper selectable: 5 × 7 dot matrix
- 128 printable ASCII characters, 52 printable five-level characters
- Jumper-selectable RS-232-C or 20 mA loop output
- Movable ASCII cursor: home (HOM), back space (BS), horizontal tab (HT), vertical tab (VT), line feed (LF), carriage return (CR), form feed (FF), end of screen (EOS), end of line (EOL) and delete (DEL)
- ASCII mode absolute and relative cursor addressing
- Device control by means of the DC3 control character
- Five-level mode control characters: carriage return (CR), line feed (LF), letters (LETS) and figures (FIGS)

Overall view of my 1802 system. At left are memory and the associated power supply; in the center is the CPU unit; at right are the ASCII keyboard and monitor. The Netronics video board is mounted beneath the keyboard. *(Photos by James Lucas)*

The printed circuit card is high-quality glass epoxy with plated-through holes. There are no handling difficulties with the integrated circuits, most of which are 74 LS series. The video display board is designed to interface with any computer using an RS-232-C or 20 mA serial port. With a modem, it could be used as a telephone terminal.

The assembly manual includes the patches necessary to use the video board with the Netronics version of 1802 Tiny BASIC. The modifications to the software make available to the Elf II user an extra 1000 or so

bytes of memory space. The improvement in the video display over the character generator of Tiny BASIC is striking. The display is more professional in appearance, and program readouts are much easier to design. Although Tiny BASIC's PLOT command no longer functions (the 1861 video chip is not used after the program modifications), the absolute and relative cursor sequence can be used to program a substitute plot function.

Assembly

The kit is not difficult to assemble, but kit-building experience is helpful. Included with the kit are sockets for the F-8 and the character generator IC. I used sockets for all ICs to facilitate troubleshooting and repair.

The kit includes an on-board 5 V regulator. A filtered power supply is required to power the video board. If you select the RS-232-C output option, a negative supply will also be needed. The ± 8 V supply (see Fig. 1) will satisfy both needs.

Connection to the video board from an encoded ASCII keyboard is accomplished at location J-2. A 14-pin socket is provided for plug-in convenience. The Netronics keyboard connects directly to this J-2 socket. To interface to other keyboards, see Table 1 for the pin-out at J-2. Note that the strobe (or data ready) line at pin 6 can be either active low or active high.

Netronics recommends the RS-232-C output option for the Elf, so that is the one I chose. Output hookup to the Giant Board's serial port, including Giant Board jumper requirements, is covered in the assembly manual. You should ignore conflicting statements in the Giant Board manual, as the Tiny BASIC modifications correct for the differences. As modified, Tiny will communicate both with the hex keypad and the video board. You should test the assembled video board according to the manual's instructions before connecting it to your Elf. You can then patch Tiny and test the Elf and display together. This way, you will know whether a problem is in the display hardware or an incorrect patch.

ASCII parallel output	Video board J-2 pin-out
D_0	11
D_1	4
D_2	12
D_3	3
D_4	13
D_5	2
D_6	14
Strobe/ data ready	6

(ASCII D_7 is ignored)

Table 1. Video board J-2 pin-out.

Operation

While Tiny BASIC is operating, nothing will happen on the screen that does not come from the Elf. Whatever is typed on the ASCII keyboard is echoed by Tiny for display on the screen. The changes in Tiny BASIC alter some of the program's running characteristics. Print statements can no longer be frozen by depressing the input switch on the hex keypad. Breaks in program execution can still be produced by depressing an ASCII keyboard key, but a more reliable break occurs when input is used. With the 1861 chip disabled, programs run a little faster.

A substitute plot function can be programmed using the cursor sequence as described in the manual. There is also a downshift sequence allowing display of special print characters. These are mostly mathematics symbols and portions of the Greek alphabet.

User Comment

My setup uses a portable television set with rf modulator. Netronics recommends selecting the 32-character line with this arrangement. With my TV-1 modulator, the 32-character display line is excellent. My preference is for the 64-character line, which is still quite readable. The higher density line is better for programs using graphic displays.

Video board operation differs from that described in the assembly manual. The manual states that characters in a print statement are lost after a form feed (FF). I did not find this to be so; nothing is lost. The manual also says up to 48 characters are lost after an end of line (EOL) erase at 300 baud. In fact, 50 characters are lost for a full line erase. Proportionately less is lost for less than a full line erase, and the loss at 110 baud is also proportionately less. The end of screen erase (EOS) works

ASCII	decimal	hex	H	V	ASCII	decimal	hex	H	V
·	96	60	32	0	@	64	40	0	0
a	97	61	33	1	A	65	41	1	1
b	98	62	34	2	B	66	42	2	2
c	99	63	35	3	C	67	43	3	3
d	100	64	36	4	D	68	44	4	4
e	101	65	37	5	E	69	45	5	5
f	102	66	38	6	F	70	46	6	6
g	103	67	39	7	G	71	47	7	7
h	104	68	40	8	H	72	48	8	8
i	105	69	41	9	I	73	49	9	9
j	106	6A	42	10	J	74	4A	10	10
k	107	6B	43	11	K	75	4B	11	11
l	108	6C	44	12	L	76	4C	12	12
m	109	6D	45	13	M	77	4D	13	13
n	110	6E	46	14	N	78	4E	14	14
o	111	6F	47	15	O	79	4F	15	15
p	112	70	48	0	P	80	50	16	0
q	113	71	49	1	Q	81	51	17	1
r	114	72	50	2	R	82	52	18	2
s	115	73	51	3	S	83	53	19	3
t	116	74	52	4	T	84	54	20	4
u	117	75	53	5	U	85	55	21	5
v	118	76	54	6	V	86	56	22	6
w	119	77	55	7	W	87	57	23	7
x	120	78	56	8	X	88	58	24	8
y	121	79	57	9	Y	89	59	25	9
z	122	7A	58	10	Z	90	5A	26	10
{	123	7B	59	11	[	91	5B	27	11
:	124	7C	60	12	\	92	5C	28	12
}	125	7D	61	13	]	93	5D	29	13
~	126	7E	62	14	>	62	3E	30	14 *
?	63	3F	63	15 *		95	5F	31	15

* Substitutions

Table 2. ASCII characters as cursor addresses.

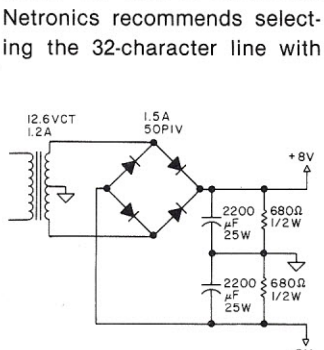
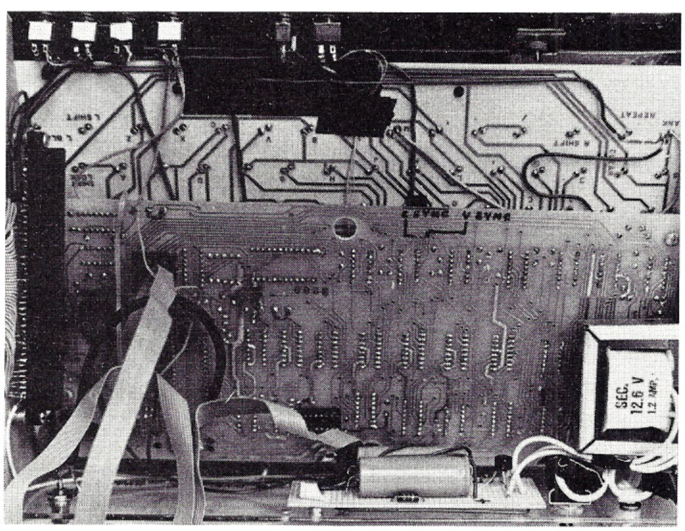


Fig. 1. ± 8 volt filtered power supply.



Video board close-up. At bottom right is the recommended power supply of Fig. 1. The connector at left is for the Netronics Giant Board I/O module, to which the video board connects.

```
0010  GO TO 50
0011  PR "[DS] A"
0012  RETURN
0050  ....
....
```

(Do not insert any spaces into the program)

Listing 1. Downshift delete print subroutine.

Program listing	Address Hex	Address Decimal	Memory (hex)
0010 GO TO 50	0B87	2951	00 0A 47 4F 54 4F 35 30 0D
0011 PR "[ESC] A A A ";	0B90	2960	00 0B 50 52 22 1B 41 41 41
			22 3B 0D
0012 RETURN			00 0C 52 45 54 55 52 4E 0D
0050 ...			
....			

(Do not insert any spaces into this program)

Table 4. Cursor address program.

as described in the manual.

Several Tiny BASIC peculiarities also cause the video display to work unexpectedly. Tiny filters out 13, 00, 0A and 7F hex characters in the input line buffers. This means that nulls (00) cannot be used as fillers for the erase functions. Use 01 hex (SOH, start of heading) or any other non-printable, non-cursor control character as filler instead.

You also cannot use delete (DEL), 7F hex, in a cursor sequence or a downshift special character print sequence. A question mark (?), hex 3F, can be substituted in the cursor sequence. If your Tiny uses hex 5E (↑) as the line cancel code, it can't be used in a cursor sequence; use >, hex 3E, instead (see Table 2). Hex 5E is not used in any downshift print sequence.

This still leaves you without the ability to print the special delete print character . . . except by trickery. While Tiny will not accept and store in program memory 7F hex, you can use a POKE statement to do the same thing. However, you must know at what address to poke 7F. Tiny BASIC, as modified for video board use, begins user programs at 0B87 hex, 2951.

Line 10 in Listing 1 sends the program to line 50, where the normal beginning of the program occurs. Lines 11 and 12 contain a subroutine that can be called from anywhere in the program. The reason for putting the subroutine so close to the beginning is to make it easier to determine where to poke 7F.

Address Hex	Address Decimal	Memory (hex)
0B87	2951	00 0A 47 4F 54 4F 35 30 0D
0B90	2960	00 0B 50 52 22 10 41 22 0D
		00 0C 52 45 54 55 52 4E 0D

Table 3. Hex memory dump of Listing 1.

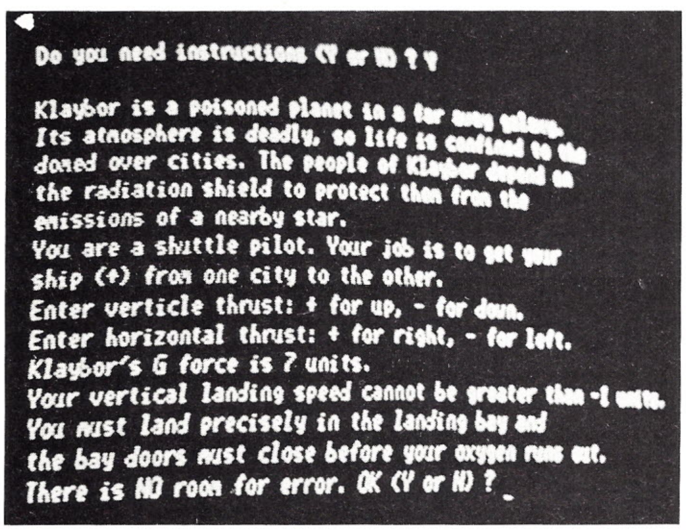
Table 3 shows a hex memory dump of Listing 1. You can see that the technique is to type the Listing 1 program, using A as a filler, and then to POKE location 0B96 hex, 2966, with 7F hex, 127. The POKE command can be part of the main program or executed directly.

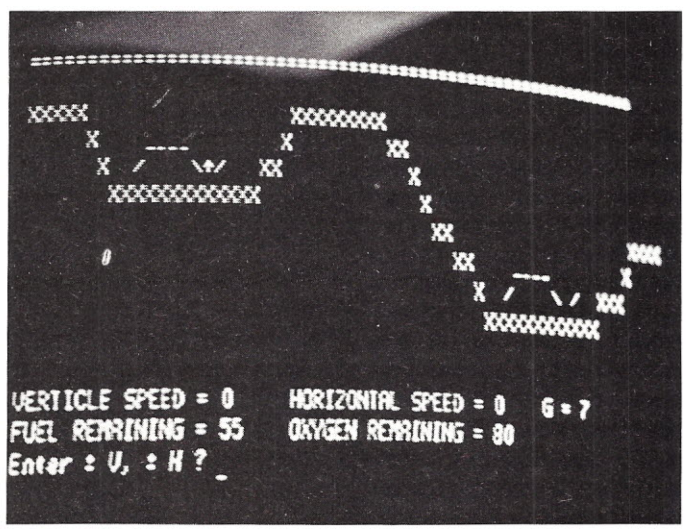
You now have a subroutine to print the special downshift delete character. The extra effort is necessary because this character is useful in creating graphics. You can also use this technique if your program needs to print 0A hex (line feed) or 13 hex (device control 3).

Using the video board poses three disadvantages. First, there is no built-in graphics capability as with some other computer systems. Second, if you list a program containing cursor moves, the list may write over itself with the cursor obeying the programmed commands. Third, the cursor addressing method is rigid and inconvenient as designed. This is because the data used to move the cursor is the hex code of characters (evaluated modulo 16 for vertical and modulo 64 for horizontal) rather than the value of variables. The result is that one program line is needed for each possible cursor move. Unfortunately, relative and absolute addressing both work in the same way.

While I have no suggestions to offer concerning the first two disadvantages, you can overcome the third disadvantage with a bit of trickery once again. You can use the same trick you used with the downshift sequence (see Table 4).

As before, the normal beginning of the program is at line 50. The subroutine of lines 11 and 12 can be called from anywhere in the program. Poke location 0B96 hex, 2966, with = or + for absolute or relative addressing, respectively. The hex character for = is 3D; 2B for +. Poke loca-





Examples of the kind of display obtainable using the 64-character format and an rf modulator. Netronics suggests the 32-character format with rf modulators. I had to adjust TV-1's channel and video level pots and the television's fine-tuning, brightness and contrast controls to get a readable 64-character display.

```
              Horizontal
          0 1 2 3 4 5 . . . . . . . 63
        0 x
V
        1
e
        2
r
        3
t
        4
i
        5 y
c
        .
a
        .              Examples:
l             Position          Address
        .        x (upper left)    = 0,0
        .        y                 = 5,0
       15        z (lower right)   = 15,63
```

Table 5. Cursor address screen layout.

tion 0B97 hex, 2967, with the desired value for the vertical address component, which can be the value of a variable. Poke location 0B98 hex, 2968, with the horizontal address component, which can also be the value of a variable. The vertical address component is evaluated modulo 16, while the horizontal component is evaluated modulo 64.

Refer to Table 5 for the layout of the screen according to the vertical and horizontal addresses used. This subroutine/poke technique is flexible enough to use the ASCII characters as discussed in the assembly manual. Table 2 shows the equivalent of each ASCII character used as vertical and horizontal addresses. This figure includes the substitutions discussed earlier, even though they are not required when a POKE is used. Note that any values between 0 and 255, except 13 (0D hex), can be poked. Tiny would read the 0D as a carriage return and terminate the line. A program error would result.

The print and cursor subroutines can be used together with an adjustment in program line numbers and poke addresses used.

I have two suggestions:

Install a double-pole, double-throw switch as shown in Fig. 2 between the video board and Giant Board. Thrown one way, the switch permits normal operation between the display and your Elf. Thrown the other way, Elf II is cut off from the type on the ASCII keyboard. The video board echoes to itself, so the type still shows up on the screen. This means you can, for example, clear the screen by typing form feed (FF) without fouling up your next input or program line to Tiny BASIC.

Keep your keyboard's parallel output connected to the parallel port on the Giant Board. That way, you still have use of the INP(7) function.

Summary

Netronic's video board is a good value for the money. I especially recommend it for Elf II users. It will make you feel more like you have a "real" computer setup.

Some of the problems associated with the video board can be overcome as I have suggested. Others are either insignificant or the result of Tiny BASIC quirks, which will probably disappear once a full BASIC is released. Netronics states that their full BASIC will require the video board.

As with Tiny BASIC, use of the video board will reduce the amount of memory that might otherwise be required. Netronic's full BASIC will execute out of 8K of memory, whereas Quest's full BASIC needs a little over 11K of memory. This is another good reason to consider adding the video board to your Elf.

I have written some Tiny BASIC programs (for example, a real-time lander with textual display and a flight program with a graphics display) that demonstrate the operation of the cursor address sequence. Any interested readers can send me a stamped, self-addressed envelope for more information. ■

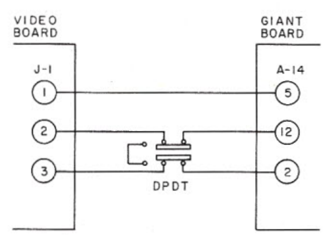


Fig. 2. Terminal isolating switch wiring.

Bluebird's Inc. ✓175

SOFTWARE for the TRS-80™

Compress-IT .... REQUIRES DISK BASIC & ANY RAM SIZE $24.95
REMOVE UNNECESSARY SPACES & REMARK STATEMENTS IN A PROGRAM. COMPRESSES A PROGRAM INTO MAXIMUM MULTIPLE STATEMENT LINES. REDUCE PROGRAM RAM REQUIREMENTS TO A MINIMUM. CREATE FASTER RUN-TIME PROGRAMS FROM A WELL DOCUMENTED SOURCE PROGRAM. COMPRESS-IT CONSISTS OF 2 PROGRAMS.

Simplify-IT .... REQUIRES DISK BASIC & ANY RAM SIZE $24.95
BREAK DOWN A COMPLEX PROGRAM & LIST PROGRAM IN SINGLE STATEMENT LINES. TITLES & NUMBERS EACH PAGE IN 11 INCH FORMS LENGTH. ELIMINATES CHOPPED OFF LINES INHERENT WITH SOME PRINTERS. SEARCHES & FINDS REQUESTED TEXT OR COMMANDS-STATEMENTS-ETC. LISTS ALL VARIABLES AND DEFINITIONS USED IN A PROGRAM. WORKS WITH OR WITHOUT A LINE-PRINTER. SIMPLIFY-IT CONSISTS OF 3 PROGRAMS.

TEXT .... REQUIRES DISK BASIC & 32K MIN. RAM $49.95
A COMBINATION TEXT PROCESSOR & ELECTRIC FILING CABINET. EXECUTIVE MAINTAINS DIRECTORY FILES AND PREVENTS FILE DESTRUCTION. EVEN IF THE TEXT-FILE IS GIVEN THE SAME NAME AS A EXISTING FILE. AUTOMATIC BACKUP OPTION. EXTENSIVE EDIT & TEXT BLOCK MOVE OPTIONS. FAST SEARCHES THROUGH EVERY FILE EVER WRITTEN WITH PROVISION FOR CHANGING DISKS DURING SEARCH. FORMATS & PRINTS TEXT FILES TO USER SPECIFICATIONS. WORKS WITH 1 TO 4 DISK DRIVES.

List 'n File: Names & Things REQUIRES DISK BASIC & 32K MIN. RAM $34.95
CREATE & MAINTAIN MULTIPLE CATEGORY FILES. CATEGORY FILES ARE USER DEFINED AND AUTOMATICALLY MAINTAINED. ALLOWS ANY KIND OF INFORMATION TO BE EASILY FILED. EXTENSIVE FILE EDITING CAPABILITIES. EXTENSIVE FILE SEARCH OPTIONS. PRINTS LABELS OR DATA LISTINGS OF ALL FILES OR SEARCHES-FOR AND PRINTS ONLY FILES CONTAINING USER SPECIFIED KEY WORDS.

Compare .... REQUIRES DISK BASIC & ANY RAM SIZE $14.95
COMPARE ANY TWO PROGRAMS. UTILITY LISTS DIFFERENCES IN THE PROGRAMS TO VIDEO AND/OR LINE-PRINTER. A VERY USEFUL TOOL TO EVALUATE FREQUENTLY UPDATED PROGRAMS.

DISK .... REQUIRES DISK BASIC & 32K MIN. RAM $14.95
THE HOW-TO RUDEMENTS OF YOUR DISK SYSTEM. INTERACTIVE WITH LOTS OF DYNAMIC EXAMPLES. HELPS A BEGINNER USE THE DISK SYSTEM. RECOMMENDED FOR USE IN SCHOOLS.

Math/Stat Pac .... LEVEL 2 TAPE OR DISK SYSTEM & ANY RAM SIZE $39.95
LINEAR - MULTIPLE - POLYNOMIAL - GEOMETRICAL - EXPONENTIAL REGRESSION ANALYSIS. SIMULTANEOUS EQUATIONS. BARGRAPH WITH AUTO-AXIS. PLOT ANY EQUATION OVER ANY RANGE (REDUCE OR ENLARGE) WITH AUTO-AXIS. STATDAT DATA BASE ALLOW ENTRY AND EDITING OF DATA WHICH CAN BE STORED ON TAPE OR DISK. STATDAT FILES ARE READ BY THE ANALYSIS PROGRAMS. A PACKAGE OF INTERESTING EQUATIONS IS INCLUDED FOR USE BY THE GRAPHIC PROGRAMS. MATH/STAT PAC CONSISTS OF 9 PROGRAMS.

ALL BLUEBIRD'S PROGRAMS ARE IMMEDIATELY AVAILABLE, AND ARE SHIPPED WITHIN 1-WEEK OF RECEIPT. PROGRAMS WILL BE SHIPPED ON TAPE CASSETTE UNLESS YOU REQUEST A DISK COPY. NOTE: PROGRAMS ARE NOT SUPPLIED ON A DISKETTE WITH TRS-DOS. DISK COPIES ARE ON DATA-DISKETTE ONLY. CREDIT CARD ORDERS MAY BE PLACED BY PHONE: (313) 285-4455

** BLUEBIRD'S NEW FAST SERVICE ADDRESS **

VISA — master charge

2267 23rd STREET
WYANDOTTE, MI. 48192
313-285-4455

3G LIGHT PEN — PET APPLE TRS-80 (Level II)

✓241

- **Bypass the keyboard** and interact directly with the screen. Makes your computer more versatile for the novice as well as the more sophisticated user.
- The use of the 3-G Light Pen is limited only by your imagination! Use it to experiment with **graphics**, display a menu for quick **data retrieval**, as a **teaching tool** with your child, to proportion **recipes** instantly, or to play unique **games**.
- At a major medical center an anesthesiologist uses our pen to select proper dosages. In Holland they use it to create graphics. A man in New York uses it to teach his pre-school age daughter how to match capital with lower case letters. Teachers are using it in science and foreign language classes.
- **NO ASSEMBLY NECESSARY. READY TO PLUG IN.** Detailed sample programs included. Complete documentation so you can write your own programs in **BASIC**. No machine language coding necessary. All Professional models plug into machine ports and don't require batteries. Economy model plugs into tape recorder and batteries are included.
- **YOU GET:** 1) 3-G Light Pen
  2)* Demo-Game Cassette (with Professional TRS-80, PET and Apple)
  3) Sample Program.
  4) Complete documentation and instructions.

MANY GAMES AND OTHER LIGHT PEN SOFTWARE AVAILABLE.

——MAIL COUPON OR CALL TODAY FOR IMMEDIATE DELIVERY——

3G COMPANY, INC. DEPT. KB ✓241 **(503) 662-4492**
RT. 3, BOX 28A, GASTON, OR 97119

☐ TRS-80 Economy	☐ TRS-80 Professional	☐ PET Professional	☐ Apple Professional
$19.95	$34.95	$31.95	$32.95

Yes, I want to make my computer more versatile. Rush me ______ 3-G Light Pens. (Add $1.50 for mailing and handling - $6.00 foreign.)

Enclosed is: ☐ check or money order ☐ Master Charge ☐ Visa

Card No. ______ Exp. date ______

NAME ______

ADDRESS ______

CITY ______ STATE ______ ZIP ______

REMEMBER, 3G OFFERS A 30 DAY UNCONDITIONAL MONEY BACK GUARANTEE!

WHY LOWERCASE?

You can't appreciate just how NECESSARY dualcase is on your TRS-8Ø until you've seen it. Once you use dualcase you'll never want to go back to UPPERCASE only again. In fact, you'll wonder how you managed without it for so long.

The character generator in your TRS-8Ø ALREADY contains the following; greek characters, numbers, special symbols, PLUS both UPPER and lowercase letter sets. Block graphics are unaffected. Wouldn't you like access to YOUR entire typeset? Level II Basic converts lowercase command words into UPPERCASE. All characters contained between quotes remain as typed, but the software in an unconverted TRS-8Ø allows UPPERCASE display only! This software shortcut allowed Tandy to omit one video memory chip. This chip must be added and the video software repaired before the display of dualcase is possible.

Unfortunately, converting your TRS-8Ø requires installing the video memory chip plus wiring changes. There is only one modification on the market which eliminates most of the wiring. To get the dualcase mod installed you have three choices: 1) Send your computer to a company or individual who will do the wiring, 2) do it yourself, or 3) "THE PATCH"

To make choices 1 & 2 operate requires using software overhead in the form of a "driver". This takes 3Ø bytes, unless you want a "normal" shift to UPPERCASE keyboard. That takes upwards of 6Ø more bytes. Software oriented mods have three more disadvantages: 1) They reside in program memory, eating program space which you could be using, 2) other machine language programs are unusable if they are loaded against the top of memory, or 3) the "driver" software MUST be loaded every time you power-up, or the "MEMORY SIZE?" appears due to program bomb. Choice number three suffers from NONE of the software overhead problems. We call it "THE PATCH" and it's new for the 8Ø's!

"THE PATCH", a small electronic module which plugs into the unused ROM socket on Level II machines, makes necessary software changes to ROM supporting lowercase, an optional block cursor, & extra keyboard debounce. Electronically means NO software overhead. Your computer displays lowercase instantly upon power-up, and the keyboard operates in "normal" typewriter fashion.

"THE PATCH" is completely compatible with your TRS-8Ø since it is the first, and only, TRS-8Ø lowercase system designed that flawlessly mates with the computer as a unit, not just a special program package.

"THE PATCH" is also the only modification of any kind which can have extra options and updates factory installed for 5 to 1Ø dollars per option, as they are available. Same day turnaround.

Order yours today to avoid extra delay. Send $59.97 + 2.5Ø for S&H to: CECDAT "THE PATCH" / Box 8963 / Moscow, ID 83843.

THE PATCH is a Trademark of CECDAT, Inc. ✓85

master charge — VISA

EDUCATION AND INTRIGUE ON THE TRS-80

Children's Educational Packages

Playful Professor Complete step-by-step tutoring in integer mathematics and fractions using a highly graphic game format. A 60 room castle, intelligent ghost, two player option, difficulty selection, and ease of use have made this our most popular program. All ages. $9.95.

Money Master Teaches the proper counting of money, both paper and coins. A mini-adventure where rooms, creatures, and objects are represented graphically requires the child to properly pay for treasures and tolls as well as count his change. All ages. $9.95.

Intrigue and Adventure

Ghost's Gallery The popularity of the Playful Professor with adults prompted us to remove the problem portion and add even better graphics, an even smarter ghost, secret passages and the dreaded Ghost's Gallery. All ages. $9.95.

Human Adventure An adventure using the human body as a unique maze. Teaches anatomy and function. Allows either simple exploration or a race against time to cure a patient of cancer and escape before the body destroys you. All reading ages. $9.95.

Reality Ends A full scale adventure that allows you to walk between parallel universes searching for a way to defeat the forces of The Road, who threaten the existence of reality itself. $9.95.

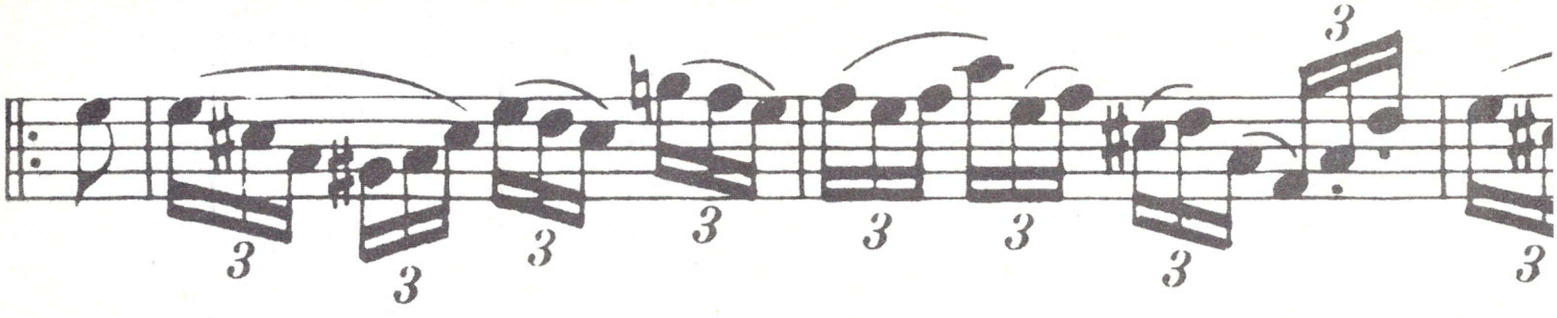
Basic Bartender Mixology data base manager. With 102 full page recipes and room for many more. $9.95.

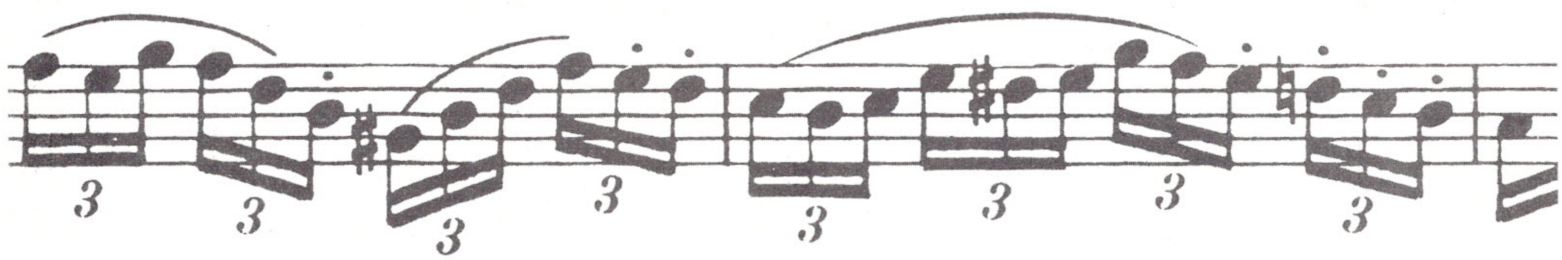
GUARANTEE: If you aren't satisfied, return your order within two weeks for a full refund.

Each program supplied on cassette with a booklet and requires 16K Level II. All orders mailed within 2 working days.

ORDER 2 AND CHOOSE A THIRD FREE!
(Sold at fine computer stores everywhere)

Med Systems Software ✓129
P.O. Box 2674, Chapel Hill, N.C. 27514



JazZ-80

Requiring an inexpensive S-100 microcomputer to control an eight-voice music board, this constructor/conductor used the Z-80 Starter System to build his music generator.

Cary N. Davids
6000 Puffer Rd.
Downers Grove IL 60515

I needed an inexpensive S-100 microcomputer to control an eight-voice music board that I designed and built. I arrange music for a 17-piece jazz band, and occasionally I need to check the flow of harmonies in short passages scored for many instruments. Since it is impractical to assemble an orchestra just for this purpose, I decided to construct a computer-controlled music generator that would enable me to hear the harmonies as I wrote them.

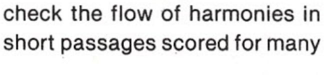


Fig. 1. Simple circuit, using existing unused inverter, which produces S-100 signal PDBIN.

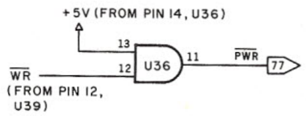


Fig. 2. Simple circuit, using existing spare AND gate, which produces S-100 signal PWR.

My requirements for the computer included: machine-language programming capability, display, keyboard, cassette interface, 2 MHz clock rate and system monitor in ROM. I was prepared to add sufficient memory and a real-time clock for my application, and I preferred to purchase the computer in kit form. After searching the ads in the major microcomputer magazines, I realized that I could assemble such a system, but it would cost more than $1000.

Not wanting to spend that much money, I let things slide until my local computer store owner showed me the SD Systems Z-80 Starter System, which is a single-board computer with a hexadecimal keyboard and six-digit display. I had originally wanted an ASCII keyboard and a video display, but after reading the Starter System brochure, I became less concerned with these features.

Hardware

The Z-80 Starter System is a complete (except for power supply) Z-80 microcomputer on a 12 inch × 12 inch board. It has an onboard keyboard and a six-digit display, 300 baud Kansas City cassette interface, wire-wrap area and a 5 V 2716/2758 PROM programmer. The system includes 1K bytes of static RAM, expandable to 2K on the board, a 2K ROM monitor and a four-channel counter-timer (Z-80-CTC). The kit version which I purchased also came with the Z-80-PIO chip, which provides two parallel I/O ports.

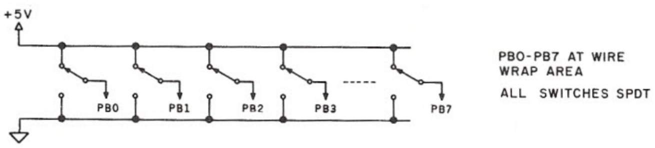


Fig. 3. Schematic of sense switch panel, interfaced to Z-80 Starter System via Z-80-PIO I/O port B.

The heavy-grade circuit board has a silk-screened parts legend and is solder masked on both sides. Address, data and control signals from the Z-80 CPU, as well as the outputs from the Z-80-CTC and Z-80-PIO, are conveniently available at the wire-wrap area. Two 24-pin sockets are provided for PROMs, one of which is also used for programming. Documentation includes a 114-page operations manual and the 30-page *Mostek Z-80 Micro-Reference Manual*. Sockets for all ICs are provided.

The feature that finally sold me, however, was the expansion area set aside for two S-100 connectors. At the kit price of $249 ($399, assembled and tested), it appeared to fulfill all my requirements.

Assembly

After unpacking all the parts and checking them off on the parts list, I began assembly. It begins with the installation of the 15 support legs, followed by the IC sockets. I recommend doing the resistors before the sockets, since they are the smallest. In this way, the resistors will be held in place properly for soldering when the board is turned over.

A careful assembly job pays off, since most Starter Kits returned to the factory suffer from soldering errors made during assembly. I put mine together in about 4 hours, with the worst part being the search for a place to put capacitor C2, which, it turned out, is an extra.

After carefully checking for solder bridges, I applied power with no ICs present (smoke test). A single +5 V dc regulated power supply capable of supplying 1 Amp is required. (A separate +25 V 30 mA supply is needed for PROM programming.)

When purchasing or building a power supply, you should keep in mind the expansion capabilities of the Z-80 Starter System and make provision for the voltages needed by S-100 boards, namely, unregulated +8 V and ± 16 V dc. Such power supplies are available for minimal investment from manufacturers advertising in the microcomputer magazines.

The appearance of the "prompt" character on the display when all ICs were plugged in and the power turned on signaled that all was OK. See Photo 1 for a view of the completed kit.

Firmware

Programming the Z-80 Starter System is done using the hexadecimal keyboard and display, along with the 2K monitor program ZBUG in ROM. Commands available in ZBUG allow the programmer to examine and change data in memory, I/O ports and 21 of the 22 CPU registers.

For debugging programs there is a single-step key, and up to five breakpoints can be inserted in a program. With the latter two capabilities, you can return control to the monitor program after executing one or more user program instructions, allowing inspection of memory, port and register contents. After the program has been debugged, it can be saved and retrieved using the cassette dump and load commands. An LED lights up when loading data from tape, making program storage on an inexpensive cassette recorder convenient and reliable.

Programming

The Z-80 CPU instruction set consists of all the 8080A instructions, plus 80 additional instructions specific to the Z-80. An important feature implemented by these new instructions and the

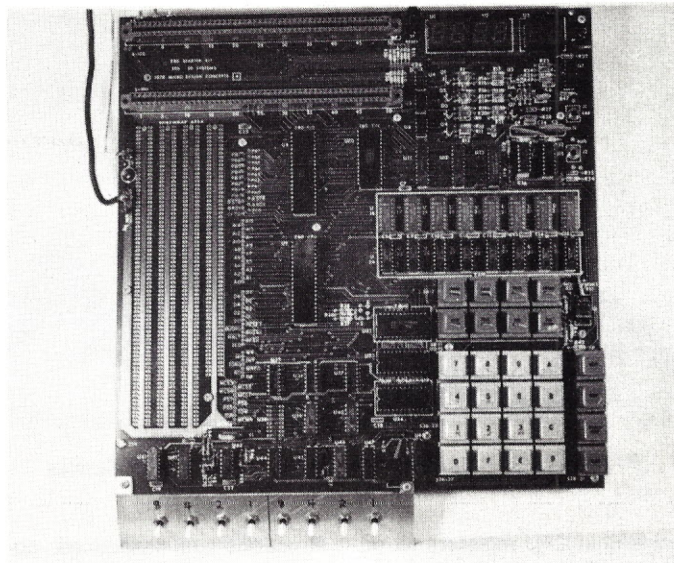


Photo 1. The completed Z-80 Starter System, including sense switches.

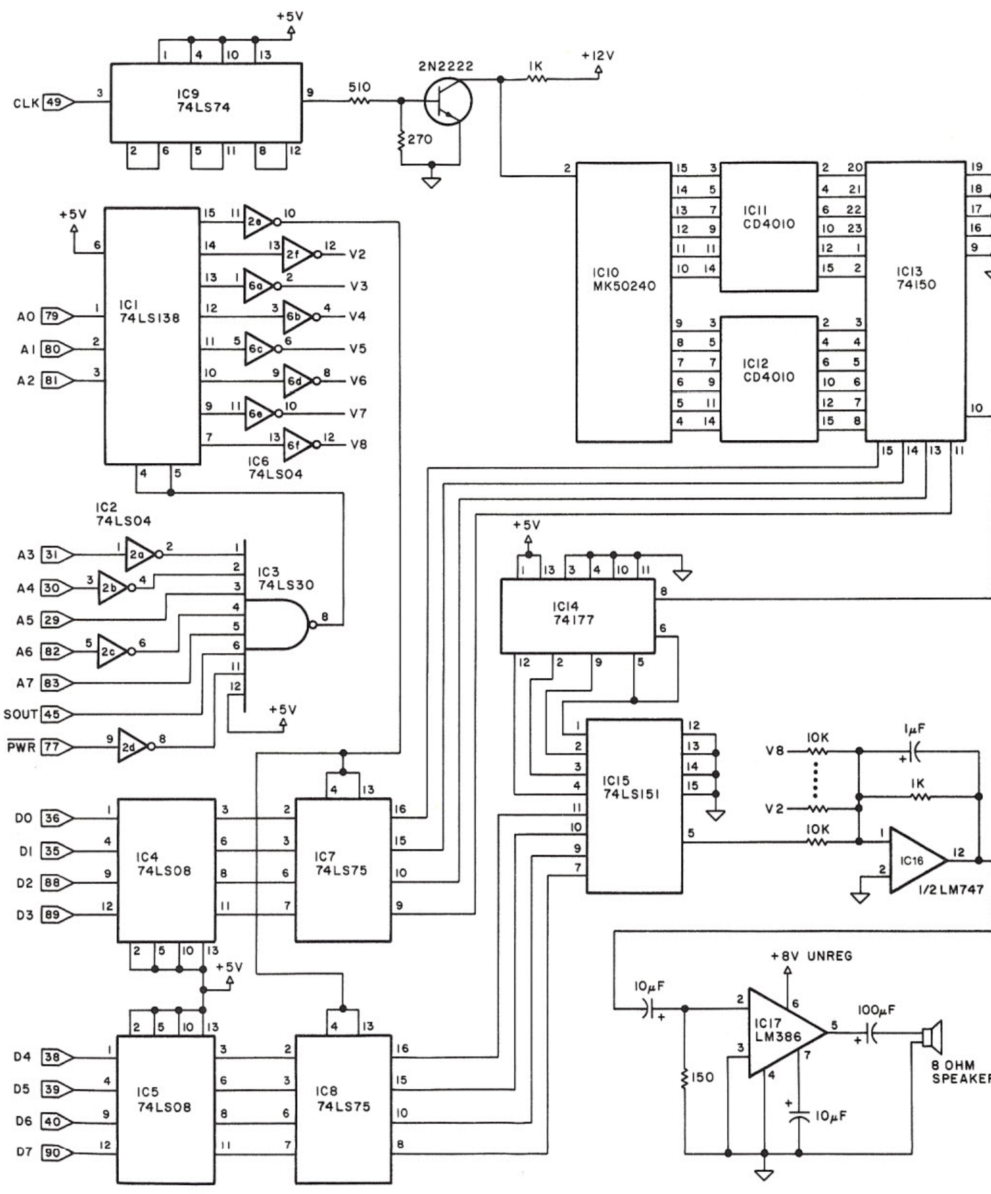


Fig. 4. Schematic of one voice and common control circuitry of eight-voice S-100 music board.

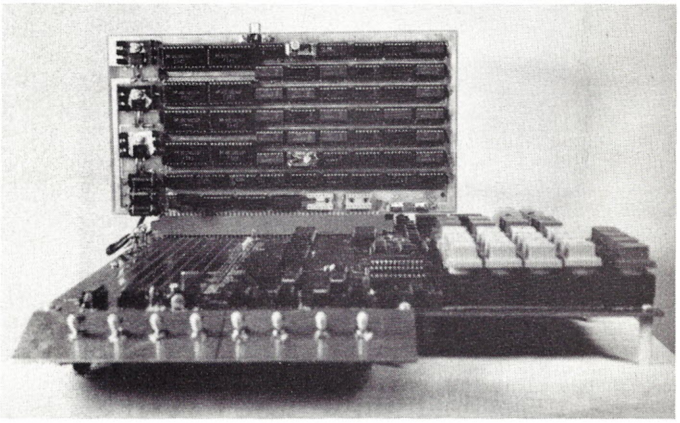


Photo 2. S-100 music board installed in Z-80 Starter System.

3000	3E	CF	LD A,0CFH	Mode word for PIO section B.
3002	D3	83	OUT(83H),A	Send to control register.
3004	3E	FF	LD A,0FFH	Control word for PIO B.
3006	D3	83	OUT(83H),A	Send to control register.
3008	DB	81	IN(A),81H	Input sense switches to A.

Table 1. Sense Switch routine.

additional CPU registers found in the Z-80 is relative addressing. In this mode a new address is determined by adding a two's complement displacement to the previous address. For example, a relative jump might refer to an instruction ten memory locations ahead in the program. The ZBUG monitor has a routine that automatically calculates these displacements and inserts them in the proper memory locations.

The advantage of relative addressing is that when these instructions are assembled, absolute addresses need not be specified. This is important for producing relocatable code; if a new instruction is to be inserted somewhere in the program, new addresses do not have to be computed for instructions utilizing relative addressing. In addition, block move instructions facilitate transfer of whole program sections in case such insertions are needed.

A powerful set of interrupt instructions is offered by the Z-80, and these are utilized by the Z-80-CTC and Z-80-PIO chips as well.

In order to take full advantage of the capabilities of the Z-80 Starter System, you should read the following two books: the *Mostek Microcomputer Z-80 Data Book* for hardware information and the *Mostek Z-80 Programming Manual* for descriptions of the instructions. Both are available for a few dollars each, and have returned this investment in time saved many times over.

S-100 Expansion

The Z-80 Starter System comes with two positions wired for Imsai-type S-100 solder tail edge connectors (not included in the kit). However, not all the standard S-100 bus signals are available at the connectors. Therefore, any desired expansion must be carefully planned in order that all the required bus signals can be made available.

I decided to aim for an additional 8K of RAM as the first S-100 addition. SD Systems recommends that only static memory be used in the Z-80 Starter System. Their 4K board evidently plugs in with no modification, but it is no longer available.

I wrote to a number of manufacturers of 8K static memory boards, requesting a schematic so that I could determine which board would interface to my system with minimum modification to the computer. As a matter of principle, I decided to make no modifications to any S-100 boards installed in my system.

The 8K static memory board offered by Digital Research Corp. (PO Box 401247, Garland TX 75040) seemed to be the answer. The only S-100 bus signal it needed that was not provided by my computer was PDBIN. Fig. 1 shows a simple circuit using an extra inverter in U37 and the Z-80 signal $\overline{RD}$, which will produce PDBIN. It requires soldering in only two jumpers. When I plugged the memory board in, a memory test program (given in the back of the *Mostek Data Book*) showed that everything was working perfectly . . . and it has continued to do so. The memory board was addressed at 4000H, which is the third 8K segment.

One additional S-100 bus signal often needed for output is $\overline{PWR}$. Fig. 2 shows how to obtain this signal using a spare AND gate in U36 and $\overline{WR}$ from the Z-80.

To aid in debugging software and new hardware, I have mounted eight SPDT sense switches on an aluminum panel and interfaced them to the Z-80-PIO parallel port B. Table 1 shows a program that initializes

Program listing.

```
Initialize the duration address table:

2000 CD A7 06              CALL UFOR4           Clear display
  03 21 29 21              LD HL, DURADTAB      Point to dur. addr. table.
  06 3E 01                 LD A, 01H            Put 1 in low byte
  08 11 02 00              LD DE,0002H          of duration addr. table
  0B 06 08                 LD B,08H             for each voice.
  0D 77              TABL: LD (HL),A
  0E 19                    ADD HL,DE
  0F 10 FC                 DJNZ,TABL-$
  11 21 2A 21              LD HL,DURADTAB+1
  14 3E 40                 LD A,40H             Put 40H,44H,48H etc
  16 06 08                 LD B,08H             in high byte of
  18 77              TABH: LD (HL),A            duration addr. table
  19 19                    ADD HL,DE            for eight voices.
  1A C6 04                 ADD A,04H
  1C 10 FA                 DJNZ,TABH-$

Set up the current duration table:

201E D9                    EXX                  Exchange registers.
  1F AF                    XOR A                Clear A.
  20 21 39 21              LD HL,NVOC           Point to number of voices.
  23 4E                    LD C,(HL)            No. of voices to C.
  24 11 21 21              LD DE,DURTAB         Point to duration table.
  27 FD 21 29 21           LD IY,DURADTAB       Point to dur. addr. table.
  2B FD 6E 00        LOOP: LD L,(IY+0)          Get duration address
  2E FD 66 01              LD H,(IY+1)          in HL.
  31 ED A0                 LDI                  Move (HL)→(DE), dec C, inc. DE,HL.
  33 B9                    CP C                 Is C=0?
  34 28 06                 JR Z, CONT-$         If yes, continue.
  36 FD 23                 INC IY               If no, keep transferring
  38 FD 23                 INC IY               durations.
  3A 18 EF                 JR LOOP-$
```

SuperBrain® Software.

	MICROSOFT	C-BASIC	PRICE
A/R	X	X	$250.00
A/P	X	X	$250.00
G/L	X	X	$250.00
P/R	X	X	$250.00
Inventory	X	X	$250.00
Restaurant Payroll	X		$250.00
Mailing List	X		$150.00
Word Processing	X		$195.00

"Industry Standard" programs on 5¼" diskette include source and complete professional documentation. Ready to run on SuperBrain.® One time charge, non exclusive license.

CMC COMPUTER MARKETING CORPORATION ✓252

116 South Mission
Wenatchee, WA 98801
(509) 663-1626 Ask for wholesale division

Also SuperBrain® computers check on prices.

® Trademark of Intertec Data Systems

GET YOURS FREE

PLEASE RUSH MY FREE 1980 CATALOG ✓106

name: ____________________
address: ____________________
city: __________ state: ______ zip: ______

PAiA ELECTRONICS, Dept. 5-K 1020 W. Wilshire, Oklahoma City, OK 73116

TORA SYSTEMS INC.

"TRS-80 is a registered trademark of TANDY CORP."

Call about our fantastic price on 4-drive complete system.

Level II 4K	$557.10
Level II 16K (w/o num. keypad)	$750.00
Expansion Interface	$269.00
Expansion Interface 16K	$403.20
Expansion Interface 32K	$524.00
16K Memory Kit for TRS-80 or Apple	$79.95

TRS-80 & NORTH STAR ADD-ON DRIVES

CUSTOM ENCLOSURE — CABLE INCLUDED

Single drive system in custom enclosure	$414.12
Single drive system in metal enclosure	$375.00
Double drive system in custom enclosure	$824.14

MPI	$279.00
MPI, B52, dual headed	$349.00
Shugart SA400	$286.00
Shugart SA800	$479.00
Tandom single sided	$279.00
Tandom double sided	$425.00
Siemen 8" drive	$430.00
Single tier walnut enclosure for Shugart	$27.00
Double tier walnut enclosure for Shugart	$45.00
Atari 400	$548.49
Atari 800	$994.49
Hazeltine 1400, 18 month warranty	$649.95
Centronics PI Printer (TRS-80 add on)	$398.95
Centronics 779-2 tractor (TRS-80 add on)	$1049.95
TI Printer	$1599.00
Base 2	$599.00
Horizon 1, 32K	$2290.00
Superbrain	$2895.00

SPECIAL! MINI FLOPPY DISKS, box of 10 (with plastic box) only $28.00 (without plastic box) only $26.50. Box of 10. 8" disks (in plastic box $30.00). Centronic 779 ribbons $3.50 each.

WE ACCEPT BANK AMERICARD, VISA, MASTER CHARGE

29-02 23RD AVENUE ASTORIA, NEW YORK 11105
(212) 728-5252 Telex #420001 ETLX

FREE ! up to $170. in merchandise with purchase of PET-CBM item ! ! !

		FREE MERCH.
PET 16K Large Keyboard	$ 995	$130
PET 32K Large Keyboard	$1295	$170
PET 8K Large Keyboard (New)	$ 795	$100
PET 2040 Dual Disk (343K)	$1295	$170
PET 2023 Printer (pres feed)	$ 849	$110
PET 2022 Printer (trac feed)	$ 995	$130

KIM-1 $159 (Add $30 for Power Supply)	SYM-1 $ 209.00
AXIOM EX-801 Printer-PET	$ 477.00
2114 L 450 ns ... 5.35 24/4.95	100/4.45
2716 EPROM (5 Volt)	39.00
6550 RAM (for 8K Pet)	12.70
PET 4 Voice Music System (KL-4M)	29.50
All Books and Software	15% OFF
Leedex Video 100 12" Monitor	119.00

ATARI — INTRODUCTORY SPECIAL

ATARI 400, Atari 800, and all Atari Modules **20% OFF.**

Heath WH-19 Terminal (fact. asm.)	770.00
Programmers Toolkit - PET ROM Utilities	44.90
PET Word Processor - Machine Language	24.00

SALE

3M "Scotch" 8" Disks	10/31.00
3M "Scotch" 5" Disks	10/31.50
Verbatim 5" Disks	10/26.50
Disk Storage Pages	10/ 3.95

Cassettes (all tapes guaranteed) Premium quality, high output low noise in 5 screw housing with labels: AGFA PE 611

C-10	10/5.95	50/25.00	100/48.00
C-30	10/7.00	50/30.00	100/57.00

Add $1 per order for UPS shipping.
Ask for 6502, TRS-80, and S-100 Product List.

A B Computers 115 E. Stump Road, Montgomeryville, PA 18936, (215) 699-8386 ✓121

Initialize the display of beat number and bar number:

```
203C 3C          CONT:  INC A              A and C still 0.
  3D 57                 LD D,A             Put 1 in beat register.
  3E 5F                 LD E,A             Put 1 in bar register.
  3F DD 21 F9 23        LD IX,DSMEM2       Dend D to display memory.
  43 CD 3C 06           CALL UFOR1
  46 7B                 LD A,E             Send E to display memory.
  47 DD 21 FB 23        LD IX,DSMEM4
  4B CD 3C 06           CALL UFOR1
```

Initialize Z80-CTC as a real-time clock:

```
204E D9                 EXX                Exchange registers.
  4F 3E 35              LD A,35H           Timer, prescale 256, no int.
  51 D3 84              OUT (84H),A        CTC/0 control word.
  53 3E 20              LD A,20H           Time constant=32D.
  55 D3 84              OUT (84H),A        CTC/0 time constant.
2057 ED 47              LDI A              Interrupt vector hi byte-20H.
  59 3E 8E              LD A, 8EH          Interrupt vector lo byte=8EH.
  5B D3 84              OUT(84H),A         Interrupt vector to CTC.
  5D 3E D5              LD A,0D5H          Counter,+ve slope, int. enable.
  5F D3 87              OUT(87H),A         CTC/3 control word.
  61 3E 0A              LD A,TEMPO         10D=fast, 24D=slow.
  63 D3 87              OUT(87H),A         CTC/3 time constant.
```

Send the first note to all voices:

```
2065 AF                 XOR A              Clear A.
  66 21 39 21           LD HL,NVOC         Point to number of voices.
  69 46                 LD B,(HL)          No. of voices in B.
  6A 0E A0              LD C,0A0H          Port for voice 1.
  6C FD 21 29 21        LD IY,DURADTAB     Point to dur. addr. table.
  70 FD 6E 00   FIRST:  LD L,(IY+0)        Point to duration.
  73 FD 66 01           LD H,(IY+1)
  76 2B                 DEC HL             Point to note.
  77 ED A3              OUTI               Output note, inc HL,dec B.
  79 B8                 CP B               Is B=0?
  7A 28 07              JR Z,DONE-$        If yes, finish.
  7C FD 23              INC IY             Next voice.
  7E FD 23              INC IY
  80 0C                 INC C              Next port.
  81 18 ED              JR FIRST-$
```

Enable interrupts and enter display routine in monitor:

```
2083 ED 5E       DONE:  IM 2               Interrupt mode 2.
  85 FB                 EI                 Enable interrupts.
  86 C3 F4 00           JP DISUP           Jump  to display.
208E 90 20                                 Interrupt vector.
```

Interrupt Service Routine

Update display, check if song is over. If so, clear all voices:

```
2090 08                 EX AF,AF'          Save registers.
  91 D9                 EXX
  92 DD E5              PUSH IX
  94 0C                 INC C              Increment interrupt counter.
  95 79                 LD A,C             Bring to A.
  96 FE 0C              CP 0CH             Is it 12?
  98 28 22              JR Z,BEAT-$        If yes, jump to BEAT.
  9A FE 18              CP 18H             Is it 24?
  9C 28 1E              JR Z,BEAT-$        If yes, jump to BEAT.
  9E FE 24              CP 24H             Is it 36?
  A0 28 1A              JR Z,BEAT-$        If yes, jump to BEAT.
  A2 FE 30              CP 30H             Is it 48?
  A4 20 1F              JR NZ,VOICE-$      If no, jump to VOICE.
  A6 0E 00              LD C,00H           Clear C, since bar is over.
  A8 1C                 INC E              Increment bar counter.
  A9 21 3A 21           LD HL,BARCT        Point to number of bars.
  AC 7E                 LD A,(HL)          Bring in number of bars.
  AD BB                 CP E               Finished song?
  AE 28 52              JR Z,QUIT-$        If yes, jump to QUIT.
  B0 7B                 LD A,E             Move bar count to displ. mem.
  B1 DD 21 FB 23        LD IX,DSMEM4
  B5 CD 3C 06           CALL UFOR1
  B8 16 01              LD D,01H           Reset beat counter.
  BA 18 01              JR BTDIS-$         Display beat number.
  BC 14          BEAT:  INC D
  BD DD 21 F9 23 BTDIS: LD IX,DSMEM2       Move beat count to displ. mem.
  C1 7A                 LD A,D
  C2 CD 3C 06           CALL UFOR1
```

this port as an input port with no handshaking. Fig. 3 shows the circuit diagram for the sense switches. The panel and the switches can be seen in position in Photos 1 and 2.

Application — Eight-Voice Music Board

The Z-80 Starter System with the Z-80-CTC counter-timer chip is admirably suited for process control applications. The example which I have implemented is an eight-voice music system, but the programming methodology can be used for any sequence timing control function. The music board, built to S-100 standards, produces the audio tones, and the computer is used only to control which notes will be heard and in what sequence. The eight-voice capability means that eight different notes can be played simultaneously, although often several of the voices will be playing the same notes but in different octaves. At any rate, it is possible to create multi-part harmony, which was my aim when I set out to build this board.

How the Hardware Generates Music

Fig. 4 shows a schematic of one voice of the music board, plus the common control circuitry. Table 2 shows the circuit pin-outs. The primary frequencies are generated as square waves by the MK50240 Top Octave Generator IC10 (see the references for more information on this method of generating audio tones with hardware). It does this by dividing the 2 MHz system clock (prescaled to 125 kHz by IC9) by 12 separate integer values, resulting in 12 chromatic tones in one octave of the equal-tempered scale.

After buffering for fan-out to eight voices occurs, the desired note is selected by multiplexer IC13, and the resulting frequency is divided by 2, 4, 8 and 16 in binary counter IC14. These notes in four different octaves are selected in multiplexer IC15 and passed onto the audio circuitry.

Each voice is addressed as an output port by the computer, with logic and decoding done in ICs 1–3. The information sent on

the data bus to each port consists of an 8-bit word. The lower four bits select 1 of 12 notes of the scale, while the upper four bits determine in which of the four octaves the desired note will be.

The range of notes available is from C#, two octaves below middle C, to high C, two octaves above middle C. This roughly corresponds to the span of notes used by a dance band.

Once the data word is passed from the CPU to the music board, it is held by the latches ICs 7 and 8 until a new note (or rest) is requested for that voice. The audio outputs of all eight voices are summed by op amp IC16 and sent to audio Amp IC17.

I constructed this eight-voice music board on a Seals wire-wrap board, and, as shown in Photo 2, every IC position available is used. Port addresses are A0–A7 hex for voices 1–8.

Music Software

The control program below is not unique to the system I have constructed, except for various entries to the ZBUG monitor and the port addresses used. The monitor program is used only to update the display, which shows the current bar number and beat number (1–4). This function could easily be eliminated or modified by the individual user.

In order to play a tune, a song table must be entered in memory. I have assigned 1K of RAM for each of the eight voices, starting at 4000H for voice 1. This is enough for approximately 100–150 bars of music if used fully. Each note entered in the song table consists of a 1-byte hexadecimal word representing the note and octave, as described previously, followed by a 1-byte hexadecimal word representing the desired duration of that note.

The octaves starting at the lowest are 8X, 9X, AX and BX, where X is the four-bit number for the note. The 12-tone scale begins at C# and ends at C natural; it uses the hexadecimal numbers 0H through BH. Thus, middle C becomes the 8-bit number 9BH, while the lowest C# is denoted by 80H. A rest (no note) is 00H. Table 3 gives the notes and their hexadecimal equivalents.

The durations are given by assigning the decimal number 12 (0CH) to a quarter note. A whole note would then be 48D (30H), and a sixteenth note would be 3D (03H). Table 4 gives the musical symbols for the common note durations and their hexadecimal representations.

The program operates as an interrupt-driven sequence timer and requires a real-time clock to generate the interrupts. This function is provided by two sections of the Z-80-CTC counter-timer. The first section, channel 0, counts system clock pulses and decrements an internal register. When zero is reached, the zero time-out pulse is used as input for the channel 3 counter. These pulses, in turn, decrement an internal register, and an interrupt is issued to the CPU when zero is reached. The combination of two counters in cascade allows a wide range of real-time clock rates.

Fig. 5 shows the jumper connection required in order to cascade the two channels. This modification does not affect the regular functions of the Z-80-CTC, as it is used by the monitor program. The two internal registers and the interrupt vector are initialized before playing a song. To change the tempo

		Gnd	+5 V	+12 V	−12 V	+8 V unreg
IC1	74LS138	8	16			
IC2,6	74LS04	7	14			
IC3	74LS30	7	14			
IC4,5	74LS08	7	14			
IC7,8	74LS75 +	12	5			
IC9	74LS74	7	14			
IC10	MKS0240	3	-	1		
IC11,12	CD4010*	8	1	16		
IC13	74150**	12	24			
IC14	74177**	7	14			
IC15	74LS151**	8	16			
IC16	(½)LM747	-	-	13	4	
IC17	LM386	4	-			6

*4 needed for eight voices
+ 16 needed for eight voices
**8 needed for eight voices

Table 2. Fig. 5 pin-outs.

```
Decrement durations for each voice, output notes, update tables:

20C5 C5          VOICE: PUSH BC            Save interrupt counter.
  C6 21 39 21           LD HL,NVOC         Point to number of voices.
  C9 7E                 LD A,(HL)          Bring number of voices.
  CA 0E A0              LD C,0A0H          Port for voice 1.
  CC FD 21 29 21        LD IY,DURADTAB     Point to duration addr. table.
  D0 DD 21 21 21        LD IX,DURTAB       Point to duration table.
  D4 DD 35 00   CHEKDUR: DEC (IX+0)        Decrement current duration.
  D7 28 0C              JR Z,OUT-$         If=0, proceed to OUT.
  D9 3D          CHKVOC: DEC A             Next voice.
  DA 28 1E              JR Z,EXIT-$        Exit if done.
  DC 0C                 INC C              Next port.
  DD FD 23              INC IY
  DF FD 23              INC IY
  E1 DD 23              INC IX             Next duration.
  E3 18 EF              JR CHEKDUR-$
  E5 FD 6E 00      OUT: LD L,(IY+0)        Point to current duration.
  E8 FD 66 01           LD H,(IY+1)
  EB 23                 INC HL             Point to note.
  EC ED A3              OUTI               Output note, inc HL, dec B.
  EE 46                 LD B,(HL)          Bring next duration to B.
  EF DD 70 00           LD (IX+0),B        Update duration table.
  F2 FD 75 00           LD (IY+0),L        Update duration addr. table
  F5 FD 74 01           LD (IY+1),H        for this voice
  F8 18 DF              JR CHKVOC-$        Next voice.

Prepare to return to main program:

20FA C1           EXIT: POP BC             Restore registers.
  FB DD E1              POP IX
  FD D9                 EXX
  FE 08                 EX AF,AF'
  FF FB                 EI
2100 ED 4D              RETI               Return from interrupt
  02 21 39 21     QUIT: LD HL,NVOC         Point to no. of voices
  05 46                 LD B,(HL)          Bring no. of voices to B.
  06 0E A0              LD C,0A0H          Port for voice 1.
  08 AF                 XOR A              Clear A.
  09 ED 79       CLEAR: OUT(C),A           Clear voice.
  0B 0C                 INC C              Next voice.
210C 10 FB              DJNZ,CLEAR-$       Loop if not finished.
  0E C1                 POP BC             Restore registers.
  0F DD E1              POP IX
  11 D9                 EXX
  12 08                 EX AF,AF'
  13 D1                 POP DE             Takes care of stack.
  14 C3 AE 00           JP MON             Jump to monitor.
```

```
Table and constant area:

2121          DURTAB: DEFS 08H        Duration table.
  29       DURADTAB: DEFS 0FH        Duration address table.
  39            NVOC: DEFB 1H         Number of voices used in song.
  3A           BARCT: DEFB 1H         Number of bars to be played + 1.

Song Table Area:
4000-43FF          Voice 1
4400-47FF          Voice 2
4800-4BFF          Voice 3
4C00-4FFF          Voice 4
5000-53FF          Voice 5
5400-57FF          Voice 6
5800-5BFF          Voice 7
5C00-5FFF          Voice 8

ZBUG monitor routines used and cross reference:

UFOR4    2000                       Clear display.
DSMEM2   203F  20BD                 Display memory buffer.
UFOR1    2043  204B  20B5  20C2     Write to display
DSMEM4   2047  20B1                 Display memory buffer.
DISUP    2086                       Display update routine.
MON      2114                       Monitor program entry point.

I/O Ports used:

        81 PIO B data register     }  Used with sense switches only
        83 PIO B control register  }
        84 CTC/0 control register
        87 CTC/3 control register
        A0-
        A7 Music board voices 1-8.
```

or speed of the music, a different number may be stored in location 2062H. Execution begins at location 2000H, after the RESET switch is pushed.

Fig. 6 shows a flowchart for the music program listing. Operation of the program centers around several tables set up in RAM. The first table contains the addresses of the current note durations, one for each voice. This allows the computer to keep track of how far each voice has progressed through its song table.

Another table in RAM has the actual current durations for each voice. Whenever an interrupt occurs—once every 1/48th of a four-beat bar—each voice's current duration is decremented by one and tested for zero.

If zero has been reached for a particular voice, it is time to get a new note from the song table, output that note to the correct port and bring the new note's duration into the current duration table. The computer scans all the voices being used at each interrupt, but actually spends most of its time in a loop in the monitor program, updating the display and checking the keyboard for input.

Since the Z-80 CPU has a dual set of registers, one set is used for the music control and the other set is used for the display routine. This minimizes stack operations and allows clock times to be stored in the CPU registers.

Location 2139H holds a number from 1 to 8, representing the number of voices which will be used to play the song, beginning with voice 1. Location 213AH holds a number that should be set to one more than the number of four-beat music bars that you wish to play.

In many songs it is common to find two notes of the same name following one another. In order to produce a gap or articulation between them, you must insert a rest of short duration. For example, the first two notes of "Yankee Doodle" are quarter notes of middle C. They would sound like a half note without the articulation. Thus, they should be entered in the song table as 9B 0B 00 01 9B 0C. Notice that the first quarter note is reduced in duration from 0C to 0B to allow for the short rest of duration 01.

Conclusion

My experience with the Z-80 Starter System from SD Systems (PO Box 28810, Dallas TX 75228) has been a positive one. The possibilities for expansion are limited only by the ingenuity of the user. For example, you could add a memory-mapped video display board and an ASCII keyboard virtually by plugging them in and modifying the monitor program.

With these and an 8K or 16K memory board, you could implement with minimal effort. In short, the Z-80 Starter System

C#	80	C#	A0
D	81	D	A1
D#	82	D#	A2
E	83	E	A3
F	84	F	A4
F#	85	F#	A5
G	86	G	86
G#	87	G#	A7
A	88	A	A8
A#	89	A#	A9
B	8A	B	AA
C	8B	C	AB
C#	90	C#	B0
D	91	D	B1
D#	92	D#	B2
E	93	E	B3
F	94	F	B4
F#	95	F#	B5
G	96	G	B6
G#	97	G#	B7
A	98	A	B8
A#	99	A#	B9
B	9A	B	BA
C	9B	C	BB

Table 3. The notes and their hexadecimal equivalents.

Note	Symbol	Hex representation
Whole note		30
Dotted half note		24
Half note		18
Dotted quarter note		12
Quarter note		0C
Dotted eighth note		09
Eighth note		06
Sixteenth note		03
Triplet quarter notes	3	08 each note
Triplet eighth notes	3	04 each note
Triplet sixteenth notes	3	02 each note

Table 4. Note symbols and their hexadecimal durations.

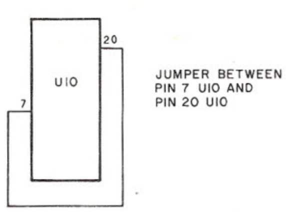


Fig. 5. Jumper required to cascade sections 0 and 3 of Z-80-CTC in order to make a real-time clock for controlling the music board.

provides an excellent way to get into microcomputing inexpensively, while still allowing an expansion capability for more advanced applications. ■

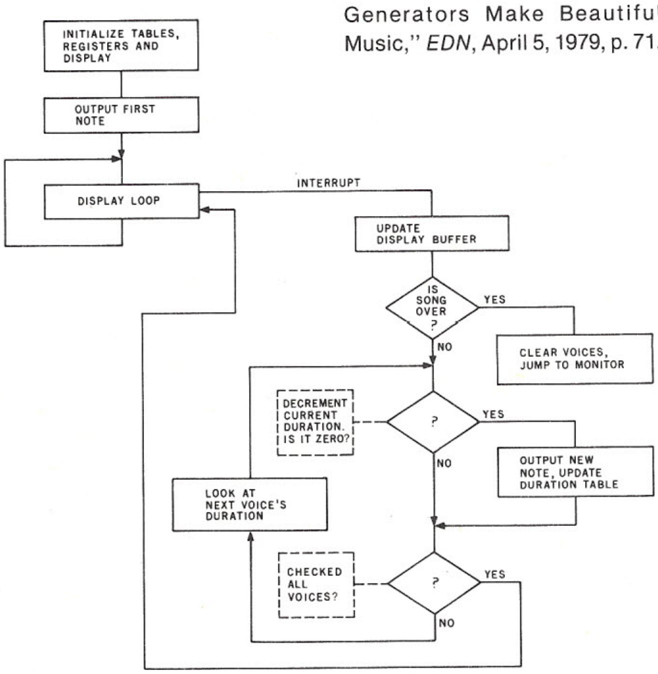


Fig. 6. Flowchart of Z-80 program used to control eight-voice music board.

References

1. Schneider, T. G., "Simple Approaches to Computer Music Synthesis," *BYTE*, October 1977, p. 140.
2. Schweber, B., "Top-Octave Generators Make Beautiful Music," *EDN*, April 5, 1979, p. 71.

Frank J. Derfler, Jr.
PO Box 17283
Montgomery AL 36117

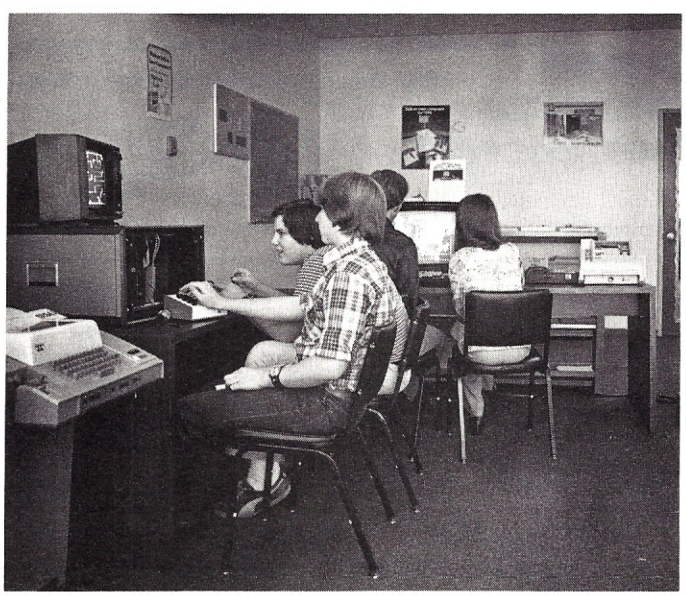
All of the Above or None of the Above?

The following is an introduction to programmed instruction. This will whet your appetite for next month's issue, which will feature microcomputers in education.

Tension was evident in the man's face. He had tried suggestion, logic, bribery (a double-dip bubble-gum sundae) and now he was ready to get physical. His nine-year-old son was fighting for the survival of a galaxy in living Compucolor. He only had a few phasor shots left, power to the shields was failing and the last Klingon was in an adjacent sector. Luckily, a final, well-placed shot ended the game and the parental conflict. The kid yelled "thanks" as he left through the door a microsecond ahead of his father.

* * *

I was in the Computer Terminal, a store in San Mateo CA. Unlike so many others, this store was located in the middle of the downtown area, not at the dark end of some backwater shopping center. It had an attractive window display, and inside were systems ready for people to use. The attitude of the clerks was "helpful" (remember how that used to be?) and the walk-in traffic of passersby, particularly preteen kids, was great. The parents were usually dragged in exhibiting the full force of well-developed sales resistance. This gradually changed to mild approval and interest until . . . somebody mentioned price. Then, with glazed eyes and fixed but drooling smiles, they dragged junior out of the store before "you break somethin' I'll have to pay for."



This is a typical scene at the Computer Terminal in San Mateo CA. There is usually a crowd of young customers working with the display computer.

The interest of kids, particularly preadolescents, in computers is natural. The powerful things that an 8- to 16-year-old can manipulate are limited. With a microcomputer, they are suddenly presented an opportunity to control an interactive TV set. They aren't particularly awed or bothered by technical complexities. The computer is simply another thing to be mastered, and they will master it by whatever rules they find.

The enthusiasm shown by the kids in the Computer Terminal was refreshing and contagious. As an educator, I knew teaching machines have been around for a long time and that computers have been used in teaching for several years. It seemed obvious that the great availability and lower cost of microcomputers could be combined with the natural interest of the students to make some effective teaching machines. The purpose of this article is to bring together various facts and experiences from the field of education and to lay them out for use by the computer hobbyist and programmer. I'm going to discuss some old and new experiments, theories and uses for computers in education. Also included is a checklist to help program writers write the most effective and complete programs possible, even with limited memory.

Some History

In the Socratic method of teaching the teacher asks a series of questions that lead the student to the desired insight or knowledge. This is probably the earliest known example of programmed instruction, that is, teaching in a step-by-step manner toward a formal goal. Programmed instruction, of course, doesn't need computers or machines; but teaching machines do need programmed instruction. A great deal of programmed instruction is in printed form, but the computer is capable of more effective interaction, flexibility and reinforcement than any printed sheet.

An early pioneer of programmed instruction was Sidney L. Pressey. His main goal was limited to giving students the immediate results

Though technically a minicomputer, this powerful Jacquard J100 is showing a pretty Hawaiian high-school sophomore some of the things micros and minis can do for education. Dual disks and plenty of internal memory give room for elegant programs, but a lot can be done with 8K and a cassette deck, too!

of tests by having them respond to test questions using a variety of electrical and mechanical devices. (If that was his goal in 1920, why does it still take a week to get an IBM sheet corrected?) His lifetime work has earned him a place as the "father" of the teaching machine. Two other later researchers worked, with a lot of pushing and pulling, to further the idea of learning by machine interaction.

B. F. Skinner is generally well known as the developer of the "Skinner Box," which taught generations of rats and pigeons many things their mothers didn't know. Skinner's method is one of careful reinforcement of the desired responses in the learner's behavior. The reinforcement is the most important focus of Skinner's work, which has, in the past, been in strong contrast to that of Norman A. Crowder, who puts emphasis on leading or coaching the learner through the material.

The differences in Crowder's work and Skinner's are mainly in approach. Despite all of the sound and fury put up by their advocates over the years, their theories are not incompatible. Let's now take a look at these theories and see how we can apply them in a practical way.

A Little Psychology

Skinner's work with animals produced a great deal of information on reinforcement, which is applicable to human beings—sort of. His experiments dealt with the relationship between: (1) the type and amount of feedback and its role in learning and (2) the way reinforcement is given and learning. Both of these are important if we are going to make our computers into effective tutors.

Research indicates that the type of feedback is not critical as long as it is positive and effective over the entire learning session. This means we don't have to make our machines sound all of their bells and whistles for every correct response, but we don't want them to just ignore a right answer, either. Flexibility and variety within a small repertoire of reinforcements are needed—a little creative programming and the flexibility of a microcomputer! The importance of the way reinforcement is given is a bit trickier to interpret. The "schedule of reinforcement" is referred to as being either "continuous" or "intermittent."

Continuous describes the case where *every* correct response made by the learner is rewarded in some way. This mode is (with good reason) the one we are the most familiar with. In intermittent reinforcement, only *some* of the correct responses are rewarded—either on a ratio basis (reward every third correct response, for instance) or on an interval basis (time). The ratio or the interval may be fixed or may vary during the training session.

Intermittent reinforcement explains why you think you have to hold your breath and cross your fingers in order to get that tape recorder to load a program into the computer. Maybe it works once every three times—you remember. Intermittent reinforcement explains a lot of what we call superstition (or does superstition explain intermittent reinforcement?).

Research shows that continuous reinforcement is effective in initially teaching some desired response. The glitch is that intermittent reinforcement seems to lead to longer retention of what is learned. Since we want the best of both worlds, it seems that the best way to provide reward for positive responses would be to first reward every correct response (continuous) and then later reward some ratio of correct responses (intermittent). This should lead to rapid learning and long retention of one learned response.

Care must be taken in applying the research on intermittent reinforcement. The effectiveness of the intermittent reinforcer only applies to repeated instances of the *same* desired response. In other words, don't reward the subject for correctly recognizing "pet" as a verb in a sentence and then ignore the correct recognition of "pet" as a noun in another sentence. Intermittent reinforcement only works for the same identical response shown repeatedly. Since (as we shall see) we will try to repeat very few correct responses, intermittent reinforcement will have only limited use for us. However, a clever programmer can make good use of this technique when repetition of correct responses does occur—perhaps in conjunction with some new learning. An example might be: RECEIVE is correct. Also, glad to see you remembered e before i (assuming that the learner had to spell out the word and had previously learned "i before e except

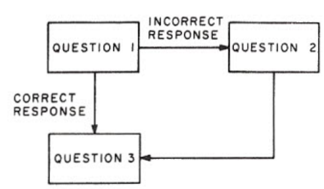


Fig. 1. Linear program format.

after c").

Shaping is another useful technique developed by Skinner. In shaping, rewards are given for actions or responses successively closer to the desired goal. In working with a computerized system, we may be better able to use the system to recognize "better and better" answers and to reward them properly. Behavior shaping is common in everyday life. It is found in dramatic and athletic coaching, peer and parent pressure and in business and political relationships.

Shaping may take up a lot of program memory, but it can be useful in teaching particularly complex tasks. Shaping may (and usually does) take place over a series of small steps or programs. It can be seen as a unifying theme among a series of programs, each of which fits into the memory available to the particular program.

Although it is unfair to treat the work of B. F. Skinner in a few short paragraphs, we must be careful about what we, as novice formal educators, can easily and practically apply. Let's take a look at the work of Crowder, the researcher whose focus is on *what is presented* rather than *how it is reinforced*.

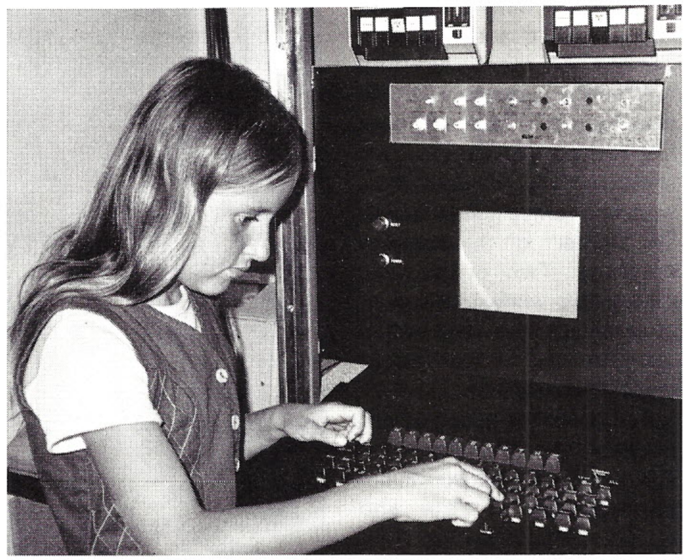
Crowder likens programmed instruction to private tutoring. He did a great deal of work with the U.S. Air Force in teaching electronic troubleshooting skills. He analyzed what a live tutor did when working with a small group of students. His results showed that the process: (1) presented new information to the student; (2) required the student to use that new information, usually to answer questions (can you think of other ways?); (3) took appropriate action in responding to what the student did (i.e., rewarding, going on to new material, reviewing in a different manner, etc.). An easy program to write, right? Yes, but . . .

Crowder's technique contains a limited amount of information and a companion multiple-choice question. The learner makes a response to the question and then proceeds to the next step indicated by the choice made. If the correct response was chosen, the next step will be additional new information; if the wrong answer is given, review and appropriate reemphasis will be given. Educators call this a "branching program," contrasted to a "linear program" (favored by Skinner), in which all learners progress through the same material in the same sequence. Internal review and smaller steps are used to insure that all learners "get" the information in the linear program format. (See Fig. 1.)

Branching learning programs appeal to most computer programmers because the learner can "flow through" the material in a more personal and individual manner. However, the technique is basically limited to multiple-choice questions, which are not always the best for several reasons that we'll discuss later. The balance is shown in Table 1.

For our purposes we should write programs to use both forms of construction. The type and complexity of the material should be the guide to how much memory to devote to fancy branching and remedial or review lessons.

Research by E. L. Thorndike and many others has shown the role of negative reinforcement in programmed instruction and in teaching machines to be very well defined—*never, not ever, nohow!* The responses "wrong,



Keyboard skills are both a part of and a valuable spin-off from computerized instruction. Many educators are calling for the development of these skills in the early elementary grades.

Branching Program	*Linear Program*
Allows individual progress.	Uses small steps and internal review to insure mastery of information.
Tries to insure all material is learned.	
Takes more memory space.	Less complex program can use many different forms of response.
Usually only uses multiple-choice questions.	

Table 1. Program comparison.

Glossary

Advance Organizers: Introductory information that can help in learning what follows.
Biofeedback: Information received by a person about his body functions. Typically concerns heart rate, respiration, brain waves, etc.
Branching Program: Allows a variety of paths through program material. Usually uses multiple-choice questions.
Computer Assisted Instruction (CAI): What the use of computerized teaching machines is called to make them less threatening to teachers.
Discovery Learning: Acquisition of new information, largely through the learner's own efforts. Good, but it takes a long time.
Frame: A unit of an instructional program. A frame is usually ended by having the learner make a response or engage in a specific activity.
Intermittent Reinforcement: Reinforcement is not provided for each repetition of the same correct response. Ratios or intervals may determine which repetitions of the response are rewarded.
Knowledge of Results: What's the score? How well did I do? For our purposes, knowledge of results should be provided immediately after each response or activity.
Negative Reinforcer: Unpleasant feedback to a response. We don't want to use it.
Programmed Instruction: Material presented in small steps or frames. Usually aimed at an overall course of instruction.
Reinforcement: Words, signs, events, symbols, etc., that take place as a result of the learner's activity or response.
Remedial Frame: In a branching program, that frame of review or reexplanation a learner might go to as a result of an incorrect response.
Response: An answer, action or activity engaged in by the learner as a part of the programmed instruction.
Schedule of Reinforcement: The manner in which reinforcement is presented (intermittent, continuous, etc.).
Shaping: A system by which complex behavior is taught by rewarding responses that are closer and closer to the one(s) desired.

JINSAM™

★CUSTOM DATA FILES
★CUSTOM REPORTS/LABELS
★KEYED RANDOM ACCESS
★FAST/EASY/MENU DRIVEN
★MULTIPLE SEARCH KEYS
★PRIVACY ACCESS CODES
★WILD CARD SEARCH

Data Manager INTRODUCTORY OFFER **$150**

JINSAM™ Data Manager for 16K-32K PET/CBM and CBM or COMPU/THINK Disk. (Printer optional). Stores up to 650 Records per disk. Has features listed above, plus

FREE: LABEL PRINTER MODULE
FREE: REPORT GENERATOR MODULE

Powerful user commands. Self explanatory, easy to use. Straight forward input and editing routines – "idiot proof". Create any desired relationship.

TYPICAL APPLICATIONS: Personnel files, Customer files, Inventory, Sales records, School records, Appointment schedules, Real estate /Apartment listings, Subscription lists, Research surveys, Mailings.

LISTED BELOW, optional MODULES which interface with **JINSAM™** to access the entire database or select Records.

MATHPACK™ allows +, −, x, ÷ on any numeric field by a Constant or other numeric field. Results temporary or permanent.

STATPACK™ descriptive statistical interface to find Average, Variance, Standard Deviation, Chi Square, Correlations, Regressions, Number of Occurances. Results to screen or printer.

WORDPACK™ Word Processor interface to personalize text by accessing field contents for mass mailings, reports, invoices . . . Text may be saved, altered and recalled. Powerful commands to edit, center, insert, delete, move blocks of text. Screen editing.

MULTI-LABEL™ module prints multiple labels per Record with 2 line caution message and consecutive numbering. Used for inventory label printing, lot numbering, serial numbering.

USER'S GUIDE only **$ 25**
DEMO TAPE $5 **DEMO DISK $ 8**
Optional MODULES (available July 1980) **$ 40** each

Specify CBM 2040 or COMPU/THINK DISK

Special Offer (Save $35)
Total Package $275
JINSAM™ + all modules above

Send Check or Money Order plus $2 Shipping
(NY residents add 8% Sales Tax)

- DEALER INQUIRIES WELCOMED -

JINI MICRO-SYSTEMS, Inc. ✓194
P.O. Box 274-K • Bronx, NY 10463

free freight

We'll pay the surface freight on all pre-paid products featured in this ad to any of 18,000 U.S. tariffed locations. No handling charges, add on costs, insurance fees or credit card fees!

Comprint 912
Low priced electrostatic matrix printer. 225 cps.
Serial **$535**
G.P. Parallel **499**

Centronics 730
100 cps, parallel, 3-way feed, 80-character buffer...the new leader in small printers. Save $100.
Call for price

Centronics 779
80 to 132 column capability, 5 X 7 matrix, 90 lines/minute (60 cps), tractors.
Call for price

TI 810
Basic 150 cps serial. The industry leader at a new low price! Save over $300!
$1589.

Soroc IQ 120
High quality, text-editing terminal. 73-key board, built-in 2K RAM.
$789.

Novation Cat
Acoustic modem with originate or answer modes for data transmission over phone lines. LED display, test mode.
Call for price

Texas Instruments 99/4 Computer
Low price includes 13" color monitor, 16-bit CPU, built-in equation calculator, TI BASIC with 13-digit precision, and more! Superior sound, color graphics. Optional solid state speech available!
Call for price

Televideo 912	$779
Centronics 704	$1685
Heath WH 14	$749

Over 2000 products from over 100 manufacturers. Prices subject to change without notice, products subject to availability. FOB Tempe, AZ (5% sales tax in Arizona)

Call now TOLL FREE 1-800-528-1418

✓156 **MICROWORLD®**
1425 W. 12th Pl. • Tempe, AZ 85281 • (602) 894-1193

dummy!" or "boy, did you blow it!" are absolutely guaranteed to create more failures than successes.

Theory

Testing is a detailed science, and it is difficult for a classroom teacher to make a good test. There are many "cookbook" recipes for writing test questions, and several things keep boiling to the top:

- Multiple-Choice Questions—easy to grade; very hard to write; useful for leading to programmed results (branching program).
- True and False—used only by mean teachers who like to trick kids. Don't use.
- Fill in the Blank—hard to write; very valuable for the student; harder for the computer to score.

Multiple-choice tests are a necessary evil. They are valuable diagnostic tools, but require a thorough knowledge of the subject and careful attention to the wording. Each multiple-choice question consists of a stem (the question) and the alternatives, two of which are foils (wrong), one is the plausible distractor (almost right) and one is the correct or keyed response. All responses should be approximately equal in length. Too many "all of the above" or "none of the above" responses are a cop-out and should be avoided. (I won't even discuss true and false questions.)

Fill-in-the-blank (completion) questions are also hard to write, but if you are careful with your words, they are easier than multiple choice. To answer a completion item, the learner must recall the total setting of the information, thereby gaining valuable practice and intrinsic reward. A completion item should be concise, unambiguous and grammatically correct. Unnecessary and excessively technical words can give hidden clues to the correct response (sometimes not bad). Caution must be observed in picking the articles and verbs used—particularly to insure that one and only one response is correct. This is perhaps the hardest part of writing multiple-choice questions.

Hard copy can serve as another form of positive reinforcement in itself. The computer can give these learners a positive report card to take home.

In writing instructional programs, as opposed to achievement tests, it is much more important for learners to get the correct answers. We want them to get the right answers and learn at the same time. The use of prompts or clues is appropriate then, and even more so in the case of completion items. Prompts can come from the context of the material. They can also come from rhymes, synonyms, antonyms and more:

- Rhyming Prompt: Jimmy Junction has three legs and a diode for a sister. Jimmy is called a ________.(transistor)
- Opposite: Heat causes things to expand. Cold causes them to ________.(contract)

Remember, it is important for learners to have a successful experience. The level of the learner and the complexity of the material must be carefully matched to ensure learning.

Discovery learning is a procedure wherein the learner is allowed to manipulate a given set of facts (numbers, figures, etc.) until he discovers their intrinsic relationship for himself. This procedure may lead to long retention of the material, particularly among younger children who are learning basic principles. It is essentially a trial-and-error method, however, and, therefore, wasteful of time and probably computer memory. We might find ways to use the discovery-learning approach, but our approach will probably be more structured and directed.

Column I	Column II
A. Reinforcement	1. Graphics
B. Presentation	2. Color
C. Scorekeeping	3. Scrolling
	4. Reverse
	5. Expanded letters
	6. Beeper
	7. Printed sheets
	8. Real objects and displays

A. 1, 2, 4, 5, 6
B. 1, 2, 3, 4, 5, 7, 8
C. 1, 2 is one set of answers. Did you have another?

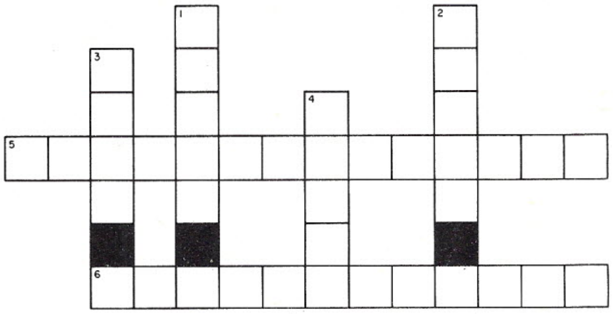
Table 2. Match the functions and capabilities. (Use as many from column II as you like.)

Brass Tacks

All that education research is fine, you say, but how do you really use it? We will now discuss how to write educational programs.

Most current teaching programs for microcomputers involve the subject of mathematics. I suspect this is due more to the inclinations of the programmers than to any feature of the computer media. Of all the elementary subjects, programmers probably remem-

. . . and if you hold this down long enough, we can all go to recess!



DOWN

1. Simulation
2. Tutor
3. Play a _____
4. Practice

ACROSS

5. If you hit five with a hammer, you would be a _____
6. Reference books.

ANSWERS

1. MODEL
2. TEACH
3. GAME
4. DRILL
5. NUMBERCRUNCHER
6. ENCYCLOPEDIA

Fig. 2. Crossword puzzle.

ber the most about mathematics; therefore, this is what they write programs about. Unfortunately, most of these programs seem to be little more than drills in mental arithmetic.

It is true that you never learn as much about a subject as when you teach it, so writing a teaching program is a beautiful opportunity to learn more about one of your other hobbies.

Many computerists are amateur radio operators. A simple primer on antenna theory could certainly take advantage of the graphics capability of some machines. The same principle applies to amateur potters, tailors, chefs, bakers and camel trainers. Some aspect of their hobbies could be taught in a step-by-step program.

Consider the following. To write a successful teaching program, you must (select one):

- A. Know your topic and provide a comprehensive description of it.
- B. Know your topic and limit it to information easily taught in small steps.
- C. Prepare your material in a firm, standardized format.
- D. Use references to expert researchers and authors in the field.

If you responded with A you should remember we need to limit the topic to what we can handle in small steps. If you responded with C or D you should remember that flexibility and creativity are the keys to computerized instruction, not standardization and expert references.

If you responded with B you are correct and are off to a good start!

The use of the available memory space in a computer calls for the most complex strategy by programmers. Writing educational programs taxes this creativity. If you have a small amount of memory available, you might put the text and questions on paper and allow the computer to do the functions of scoring, branching and reinforcing. Ideally, concrete models, displays and examples will be used to augment the printed word. The graphics available on some machines can also be a valuable teaching aid.

Probably the most important functions the computer performs as a teaching device are (select one):

- A. Presenting, scoring and teaching.
- B. Scoring, branching and reinforcing.
- C. Reinforcing, sorting and punishing.
- D. Branching, sorting and presenting.

If you chose B you were correct. What is your score now? If you were answering on a computer you would know, wouldn't you?

Here is a tougher one: Of the functions performed by a computer while teaching, the most important is probably_______.

If you didn't answer reinforcing, please go back and read the paragraphs about B. F. Skinner, but don't step into any wooden boxes.

Methods of reinforcement by computer can be written, graphic or audible. Various positive written phrases can be presented by the computer. A running tally can be kept and the score incremented graphically for each correct response (color in blocks, send a rocket to the moon, put-on-a-happy-face subroutine), and/or the end-of-line beeper can be sounded.

To further test your knowledge of teaching concepts, see Table 2.

It has been found that the sequence of steps or (to use the lingo of the educators) "frames" is not critical in shorter programs. This is good news, since our available memory will probably only allow shorter programs.

Educators call the series of steps in a program_______. (Did you say frames? Good! Now you can wow 'em at the next PTA meeting.)

Because the mind has a tendency to order material in its own manner anyway, the order in which the material is presented in shorter programs is _______. *(Not Critical)*

Combining sensitive analog input devices with our computers can give us a great capability for biofeedback

Checklist

- Select a topic and level appropriate to your learner.
- Limit the topic so it can be covered in available time and memory space.
- Divide the topic into small steps or "frames" of instruction.
- Treat each frame separately and provide an appropriate question, example or learning activity for the termination of each frame.
- Vary the terminating questions and activities to include: multiple-choice questions, fill in the blank, matching, puzzles, games, rhymes, pneumonic devices, drills.
- Provide the appropriate reward to the learner for the successful completion of each question or activity.
- Vary the feedback to include: words of praise, knowledge of results, box scores, graphics, internal rewards (as in games, puzzles, etc.), sounds, external devices (dispensing coins, tokens, bubble-gum cards, etc.).
- Use program branching as available memory and subject matter allow.
- Use other learning aids along with the computer, such as: models, examples, written text (3 x 5 cards work well), illustrations, pictures, magnetic boards, paper and pencil, flip charts, diagrams.
- Try to use "shaping" to recognize the reward responses that are close, but not quite right.
- Use "shaping" in a more general sense to gradually bring your learners up to the desired level of performance.
- Do not ever use ridicule, shame, blame or any other form of negative reinforcement.
- Don't use true and false questions. They have no class.

training. Body parameters such as skin temperature, heart rate or blood pressure could be measured and the computer used to reward periods during which the desired effect is achieved. (Here is a chance to use partial reinforcement!) The advantage of the computerized biofeedback trainer lies in its flexible reinforcement capability, patience and ability to receive and evaluate several simultaneous inputs.

As an example, imagine three people hooked up to a microcomputer loaded with a biofeedback training program. The computer prints, "Close your eyes and make alpha waves until I sound my beeper." It alternately monitors the inputs from the three analog-to-digital converters and counts to some appropriate number. After reaching a preset value in the count, it sounds its end-of-line beeper and says, "P for results." The results (time vs samples showing alpha waves) are graphed.

Educational applications of the microcomputer have all been tutorial or teaching in nature. There are other ways that the computer can be used in education. An example is the little crossword puzzle shown in Fig. 2.

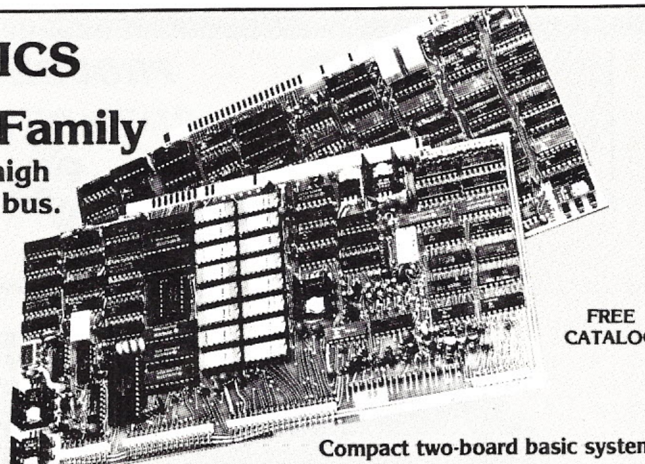
It is generally accepted that a husband can't teach his own wife to drive. Today, too, it seems as if parents can't teach their children anything they *want* them to learn. This can probably be attributed to our situation ethics and the diminished status of authority figures in our society. It is a sad but true commentary that the printed word, the computer and the television set are invested with more authority than are parents! *But* the crafty computer hobbyist, by using his computer to write words on a TV screen, can teach, preach and influence those around him. (It's fun and interesting too!) Sink your teeth into educational programs. Make them long or short, complex or simple, but use the information in the article, and the accompanying glossary and checklist to make them effective and useful.

I would be pleased to work with anyone writing "teaching" or other educational computer programs. (What are the other kinds? You don't know? Go back and work the crossword puzzle again, please.) I am particularly interested in the various kinds of reinforcements that can be developed (color, graphics, etc.), but I'd be happy to help in the formulation of questions, layout of the material, etc. Please enclose SASE. There's room for lots of creative articles and programs in this field. ■

Complete COLOR GRAPHICS With The CAT-100 Family

The original 256-color imaging system with high resolution video frame grabber for the S-100 bus.

Its digitizer captures a video frame in 1/60th of a second and stores it in its on-board image memory, with software selectable resolutions ranging from 256 to 1280 pixels per TV line. The image memory is accessible through a 2K window in the S-100 address space. The compact two-board basic system digitizes and displays 480x512 single bit pixels or 240x256 four bit pixels. The video generator displays 16 gray levels or 16 colors on standard B&W, NTSC or RGB color TV monitors.

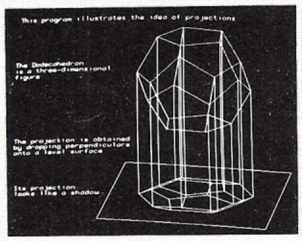


FREE CATALOG

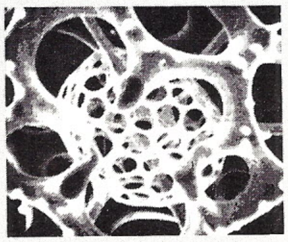
Compact two-board basic system

480x512 Contoured digitized image

240x256 Digitized image, 16 levels



480x512 Computer-generated



240x256 Digitized image, 16 levels

Features:

- Input capability from RS-170 and other TV signals
- Variety of synchronization choices
- 2 selectable video A/D conversion circuits
- Choice of 1, 2 or 4 bits per pixel
- 32K-byte image memory on the basic unit
- Expandable to 256K bytes with 24-bit color map
- Lightpen input
- Photographic trigger control input
- Character generator
- Software selectable system parameters
- Interfaces for LSI/11, TRS-80 and other processors

Accessories:

- NTSC and RGB Sony Trinitron color monitor
- High sensitivity, high resolution lightpen
- Hard copy electrostatic printer/plotter with interface
- Large variety of software packages

ROBOTICS-AUTOMATED COUNTING AND MEASURING-QUALITY CONTROL MONITORING-PATTERN RECOGNITION-IMAGE PROCESSING-MEDICAL AND SCIENTIFIC RESEARCH

DIGITAL GRAPHIC SYSTEMS ✓243
441 California Ave.,
Palo Alto, CA 94306
415/494-6088

Dept. 36

We don't say our programs are good . . .

SPACE BATTLES — High-speed graphics (machine language sub-routines). You are the commander of a small mercenary star-cruiser. You have been hired to destroy a fleet of alien invaders. If you survive, you will be handsomely rewarded; if you lose, you're dead. **cass/$14.95 disk/$19.95**

FINAL APPROACH — Land a multi-engine jet. Runway and instruments are displayed on the screen. Take control and try to land her yourself. **Special $7.50**

LEVEL IV GRAPHICS — A machine language program that allows you to turn your keyboard into a **Graphics Keyboard.** No hardware modifications required to run this program. Cassette no. compatible with Disk BASIC. **cass./$14.95 Disk/NEWDOS (not supplied) $19.95**

TOUCH TYPING I AND II — a two cassette package. Touch Typing I helps you learn the keyboard, reinforcing the correct finger-key positions. Touch Typing II randomly choses words to help you improve your typing skills. These drills can be timed or untimed. **cass./$22.50 ***

SUPER ADD — (ages 5 to adult) Improves and teaches addition skills. Divided into three categories with a total of 36 learning modules. A game can be added as a positive reinforcement tool. **cass./$24.95**

SUPER MULT — (ages 8 to adult) Contains 25 learning modules to learn and improve multiplication facts. Problems can be timed or untimed. **cass./ $24.95**

*Call in orders SPECIAL $18.95

LEVEL IV PRODUCTS, INC. ✓23

Your source for the best in TRS-80 software and hardware**

32238 Schoolcraft, Suite F4 • Livonia, Michigan 48154 • (313) 525-6200 • Out of Michigan call 1-800-521-3305

**a product of Tandy Corp.

SEND 50¢ for our new catalog!

LEVEL IV GRAPHICS — "If you write programs using graphics, this tape is a must . . . Excellent program."— 80 Software Critique, Issue #2 • "super utility program . . . excellent value — a real contribution." 80-Microcomputing, February 1980.

TOUCH TYPING I & II — "Our recommendation is to buy both programs if you are serious about improving your performance keying computer programs." — 80 Software Critique, Issue #2.

SUPER MULT — "Very comprehensive multiplication program . . . suitable for use by students just learning the multiplication tables and for adults that need some brushing up." — 80 Software Critique, Issue #2.

FINAL APPROACH — "One of the more interesting games . . . before long you'll develop a strategy . . . and end up with NICELY DONE — YOU GREASED IT ON!" — Creative Computing, January 1980.

SPACE BATTLE — "The unique value of Space Battle lies in the extra dimensions of the game . . . What is tough is staying alive . . . In short, this is a challenging game with a lot of factors to create interest . . . One of the best space war games available." — 80-US, Mar/Apr 1980.

Add $1.50 for shipping.

. . . Others say it for us!

Tarbell Disk BASIC

BASICs abound. Here's a look at how this BASIC interpreter stacks up.

Rod Hallen
State Dept.-Accra
Washington DC 20520

There are almost as many different versions of BASIC as there are types of machines to run them on. While these variations contain most of the features of the original Dartmouth BASIC, each has been enhanced by the addition of features intended to make their use easier, faster or more versatile.

Many times, while creating a BASIC program, I have felt the need for some feature or other that would simplify my task. I'm sure that this is what motivates the originators of these enhancements. Of course, these improvements do not make for compatibility among BASICs.

Over the past three years I have used the following BASICs extensively, and my opinion of Tarbell Disk BASIC is based on that use: Processor Technology's BASIC5 and Extended Cassette BASIC, Radio Shack's Level I, Level II and DOS BASIC, Microsoft's Extended Disk BASIC and BASIC-E. All of these are interpreters, except BASIC-E, which is a compiler.

An interpreter evaluates each program line as it is encountered and performs accordingly. A compiler compiles a program after it has been written, and the resultant compact machine code is actually used at RUN time. The compiled RUN code takes much less memory and runs faster than its interpreted counterpart. However, BASIC compilers are not very popular with computer hobbyists because they are much more difficult to write and debug programs with.

Since I felt that a combination interpreter/compiler would be an ideal programming tool, I tried writing and debugging programs with Microsoft or Tarbell BASIC and then compiling them with BASIC-E. This worked quite well with very simple programs, but it limited the use of the special features that each BASIC boasts about. What is needed are an interpreter and a compiler that both recognize exactly the same language.

Tarbell Disk BASIC

With this experience and these thoughts in mind, let's look at Dave Tarbell's BASIC interpreter and see how it stacks up. The one I have is designed to run under the CP/M disk operating system, but there is also a cassette version.

In order to write programs in Tarbell BASIC, you must type ENTER (or :), and from then on all entry is assumed to be program lines. A double carriage return terminates the Entry mode. Once a program is entered, it can be RUN, SAVEd or EDITed. During your first few RUNs you will probably come up with some error messages, unless the program is very small or you're very careful. More than 40 different error messages are provided, and most are self-explanatory; a few are not.

This BASIC is very space conscious. If a space is not entered before and after each reserved word (FOR, TO, GOTO, THEN, etc.), an error message will be given. I am so used to entering lines continuously without spaces that I'm still having problems. One nice feature is that most error messages are displayed as soon as a line is entered. You don't have to wait until the entire program is entered and RUN to discover that you misspelled a statement, left out an equal sign (=) or made some other simple syntax error.

The first thing that you notice about a Tarbell program is the lack of line numbers. They are not required, and I don't feel that they are necessarily desirable. In place of line numbers, Tarbell uses line designators, which are alphanumeric strings. Also, line designators are not required for every line. You can still use line numbers if you like, but why do it when words are so much more meaningful?

```
TYPE TEST
TEST     LET X=Ø:LET Y=5
         PRINT "ROD IS MY MASTER!"
         LET X=X+1
         IF X<>5 THEN GOTO TEST & 1
         PRINT
NAME     PRINT "MOST OF THE TIME!"
         LET Y=Y-1
         IF Y<>Ø THEN GOTO NAME
         END
```

Listing 1. Program written in Tarbell BASIC to show the use of line designators instead of line numbers. Note that GOTOs and GOSUBs refer to any line in the program by using a plus or minus offset from any line designator.

Operation

Listing 1 shows a simple program written to show how line designators are used. TEST and NAME are line designators. Note that GOTOs and GOSUBs can refer to a line designator or be offset plus or minus from one. GOTO TEST + 1 takes program control to the line immediately following the line designated TEST. The line designator NAME could have been left out, and then the next to the last line would read IF Y<>0 THEN GOTO TEST + 5.

Listing 2 shows another interesting feature of Tarbell BASIC: variables can also be alphanumeric. X and Y of Listing 1 have been replaced with the variables FIRST and LAST. Using words instead of letters for variables makes a program more readable because, if appropriate names are chosen, it is easier to figure out what each variable is used for.

To make changes or corrections, the EDIT mode is entered. You do this by typing EDIT and a line designator for the line you want to edit. If you wanted to change the seventh line of Listing 2, you'd type EDIT NAME + 1. That line would be displayed on the screen, and you'd have 19 single-letter commands to insert, delete, change, kill, append and more. I found the edit commands very difficult to use at first and messed up many programs until I became used to them. The major problem was that I am very familiar with the Microsoft editor and kept trying to use its commands.

Before and/or after editing occurs, you can save your program on disk. There are two methods of doing this: SAVE puts the program on disk in ASCII format, and BSAVE records in binary. ASCII files can be read and printed using the CP/M DOS, while binary files cannot. The ASCII files can also be loaded into other BASICs such as Microsoft Disk BASIC. However, that doesn't necessarily mean they will run that way. That depends on whether or not you use noncompatible statements. The advantage of binary files is that they use less disk space and load more quickly than ASCII files.

Both sequential and random access data files, which can be either binary or ASCII, are also supported. The manual doesn't go too deeply into the explanation of data files, and I haven't done a lot with them, but they appear to be much easier to handle than Microsoft's. An example inventory program that uses random access data files is given. While no documentation accompanies it, the alphanumeric line designators and variable names make it easier to read and understand.

For users interested in the mathematical aspects, the following information is given: the range of an integer number is from 0 to 9999999999 plus or minus, and the range of a floating-point number is from 9.999999999E + 99 to 9.999999999E – 99. Numbers are handled in ten-digit BCD form.

Unique Features

Table 1 lists some of the features of Tarbell BASIC that are not available in most other BASICs. Most of these are useful and well thought out. Some of them merit additional discussion.

CHANNEL, ASSIGN and DROP are part of a flexible I/O handling scheme. Up to eight logical devices and ten physical devices are supported, and their assignments can be changed at will. Each time that the CHANNEL command is entered, a map of the current I/O assignment is displayed. Table 2 shows the normal display.

The ASSIGN and DROP commands are used to change assignments. For instance, if I wanted to send output to the list device, ASSIGN 1,5 would accomplish it. Output would also continue to go to the screen. DROP 1,2 would stop screen display. ASSIGN 1,2:DROP 1,5 would shift output back to the screen.

Local variables not affected by main program operation are allowed within subroutines through the use of the GOPROC, PROCEDURE and RECEIVE statements. Variables can be passed to subroutines for local use, but they are unchanged upon return to the main program. This is a little difficult to explain, and the manual does not go into great detail on the subject. In fact, the rather thin manual does little more than give a limited explanation on any of the statements and commands, but it does recommend some good BASIC textbooks.

The RESTORE statement can be suffixed with a line designator so that the DATA pointer is set to any desired list of data. Most BASICs will only restore to the first DATA statement in the program. I have found this to be useful while using PT's Extended Cassette BASIC.

OCT, OCT$, HEX and HEX$ are used to convert from octal or hex to decimal or vice versa. MATCH will hunt through one string for the first occurrence of a second string, and SEARCH will look through a disk file for a given string.

Strong assembly-language linking commands are provided to allow the use of machine code in conjunction with BASIC programs. USR calls a machine-code subroutine with a two-byte

```
A>TYPE TEST1
TEST    LET FIRST=0:LET LAST=5
        PRINT "ROD IS MY MASTER!"
        LET FIRST=FIRST+1
        IF FIRST<>5 THEN GOTO TEST & 1
        PRINT
NAME    PRINT "MOST OF THE TIME!"
        LET LAST=LAST-1
        IF LAST<>0 THEN GOTO NAME
        END
```

Listing 2. Program of Listing 1 rewritten to show how words can be used in place of one- or two-letter variables to make a program more readable.

Statement	Purpose
APPEND	Add additional program to one or more already in memory
ASSIGN	Assigns a physical device to a logical device
CHANNEL	Displays the logical-physical device map
CHECK	Verify a program or file from the disk without actually loading it
DIR	List disk directory
DROP	Delete logical-physical device connection
GOPROC	GOTO using local variables
KILL	Delete given variable from a program
MOVEBOF	Moves to beginning of selected disk data file
MOVEEOF	Moves to end of selected disk data file
PROCEDURE	Used to declare local variables
RECEIVE	Returns local variables
RENAME	Changes the name of a disk file
SYMBOL	List all program variables and their addresses
WAIT	Reads a given input port with a specified mask until a nonzero condition results
WIDTH	Sets line width on screen or printer

Function	Purpose
CALL	Calls a machine-language subroutine
EOF	Determines if the end of a file has been reached
FILEXISTS	Determines if a stated file exists on a disk
FRE	Returns the amount of free memory space
HEX	Returns the decimal equivalent of a hex number
HEX$	Returns the hex equivalent of a decimal number
LOC	Returns the decimal address of a stated variable
MATCH	Returns the position of one string within another
OCT	Returns the decimal equivalent of an octal number
OCT$	Returns the octal equivalent of a decimal number
POS	Returns the current position of the printer
SEARCH	Look through a disk file for a given string
SPC	Prints given number of spaces on the printer

Table 1. Features of Tarbell BASIC not available in most other BASIC interpreters.

Logical device number		Physical device number	
INPUT	0	Keyboard	0
OUTPUT	1	Screen	1
LOAD	2	Cassette in	2
SAVE	3	Cassette out	3
BLOAD	4	Spare	4
BSAVE	5	Printer	5
SPARE	6	Reader	6
SPARE	7	Punch	7
		Disk in	8
		Disk out	9

CHANNELS

PHYSICAL \ LOGICAL DEVICES	0	1	2	3	4	5	6	7
0	X	·	·	·	·	·	·	·
1	·	X	·	·	·	·	·	·
2	·	·	·	·	·	·	·	·
3	·	·	·	·	·	·	·	·
4	·	·	·	·	·	·	·	·
5	·	X	·	·	·	·	·	·
6	·	·	·	·	·	·	·	·
7	·	·	·	·	·	·	·	·
8	·	·	X	·	X	·	·	·
9	·	·	·	X	·	X	·	·

Table 2. Logical and physical I/O device assignments and the map that is displayed and controlled through the use of the CHANNEL, ASSIGN and DROP commands. Input is from the keyboard; output is to the screen and printer; and all save and load operations are via the disk.

variable in register pair D-E; LOC returns the decimal address where a designated BASIC variable is stored; SYMBOL lists all program variables and their storage addresses; and a Tarbell BASIC memory map is provided in the manual so that the assembly-language programmer knows where to look for various information he might need while linking BASIC and machine code.

For those of you who still desire line-numbered programs, an auto number feature is provided. No renumber command is provided and needed since program progression proceeds in the order that the lines are entered and not necessarily in numerical order. In other words, line numbers are used like line designators, and their actual numerical value is meaningless.

APPEND is designed to load one program at the end of a previously loaded one. This can continue as long as you have memory space available. However, you have to be sure that none of the programs that are tied together in this manner utilize similar line designators. This would cause the same problems that loading two programs with the same line numbers would.

Limitations

I did run into a few problems that appear to be program bugs. DIR gives a listing of all of the files on the current disk. However, it would not work after certain types of error messages had been displayed or the ASSIGN command was used unless a CLEAR command, which would never come up on the LIST device, was given.

I earlier discussed the ability of Tarbell BASIC to save and load disk files in either ASCII or binary. This ability can result in problems unless it is used properly. Programs or files that are SAVEd in ASCII must be LOADed in ASCII; those BSAVEd in binary must be BLOADed in binary. If you attempt to LOAD a binary program, an error message will appear and you can correct your mistake. But if you try to BLOAD an ASCII program, an out-of-memory error will result and BASIC will hang up. I find it necessary to reset the microprocessor when this occurs. A practical solution seems to be to name files so that their format is evident or to use only one format in all cases.

No PRINT USING function is provided, and the manual states that subroutines using local variables are easily written to take care of this. No information on how this is accomplished is given.

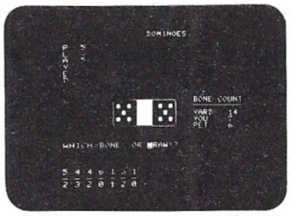
The final problem relates to the EDIT mode. Many times a program will give error messages after it has been edited, even though it appears to be correct. Completely retyping the offending area of the program banishes the error message but doesn't give any indication as to what the problem was. Apparently something remains in the program but isn't displayed on the screen.

Conclusions

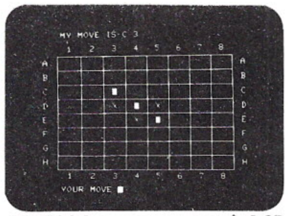
Tarbell BASIC, which is available from Tarbell Electronics, 950 Dovlen Place, Suite B, Carson CA 90746, has a lot going for it. In addition to all of the features discussed above and listed in Table 1, it will recognize most of the commands, statements and functions of the various versions of Mits and Microsoft BASIC.

And the price is right. It sells for $48, which is one-seventh of the cost of Microsoft Extended Disk BASIC. For all of you software tinkerers, the source is available on disk or paper tape for $25. All of this makes it one of the most versatile and least expensive CP/M-compatible disk BASIC interpreters on the market. ■

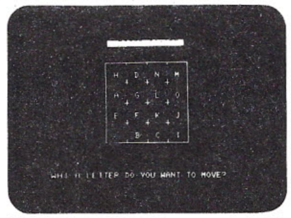
Software for the PET



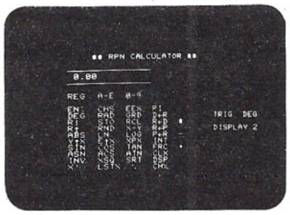
DOMINOES $ 6.95



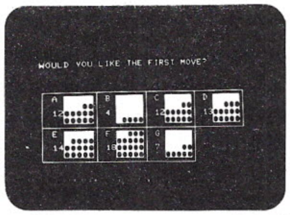
OTHELLO $ 9.95



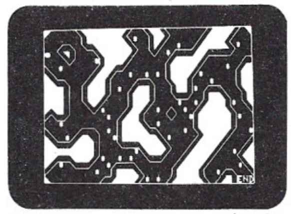
LETTER SQUARES $ 6.95



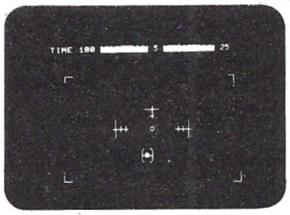
RPN MATHPACK $19.95



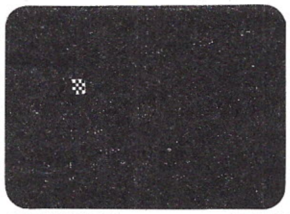
SUPER NIM $ 6.95



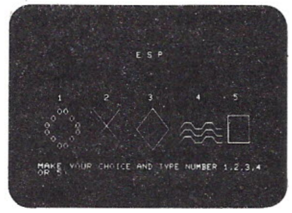
DIR/REF $ 6.95



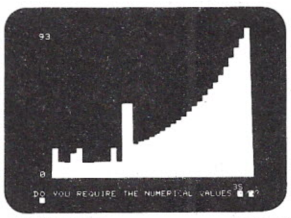
SPACE WARS $ 9.95



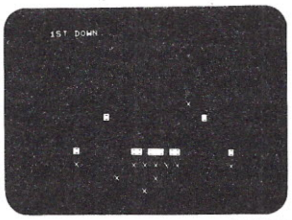
GRAND PRIX $ 6.95



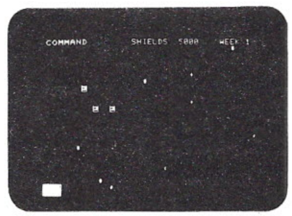
E.S.P. $ 9.95



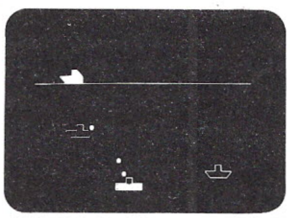
FORECAST $ 9.95



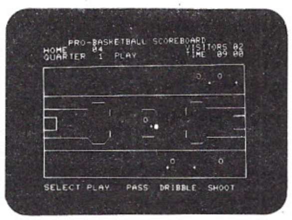
$ 6.95



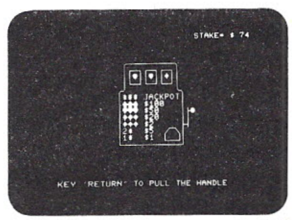
BLOCKADE $ 9.95



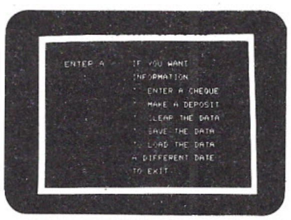
DEPTH CHARGE $ 9.95



BASKETBALL $ 9.95



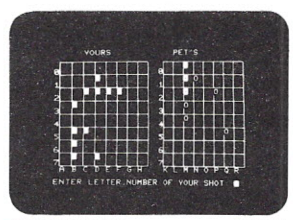
SLOT MACHINE $ 6.95



HOME ACCOUNTING $ 9.95

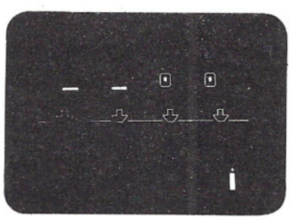
All orders include 3% postage and handling with a minimum of $1.00. California residents include 6% Sales Tax.

VISA MASTERCHARGE



BATTLE SHIP $ 9.95

PET IS A TRADEMARK OF COMMODORE BUSINESS MACHINES, INC.

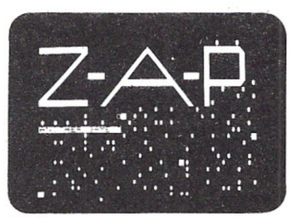


SHOOTING GALLERY $ 9.95

PROGRAMMA INTERNATIONAL, Inc.
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579
384-1116
384-1117

Dealer Inquiries Invited

✓11



ZAP $ 6.95

PROGRAMMA Software Program Products

Micromonitor for MIKBUG

An editor/assembler may be more than you need; try this 159-byte helper instead.

David L. Tietz
Route 3, 21 Rainetta Dr.
Eau Claire WI 54701

Here's a routine I've found helpful when I'm using my SWTP 6800. I have an ASR 33 TTY and mostly do machine-language programming.

When I first brought up my 6800, it took about three minutes to realize there had to be a faster, easier method of dealing with memory than MIKBUG's slow load and examine format. Waiting the 15, or so, minutes it would take my reader to crank in an Editor/Assembler was not my idea of a choice. Also the possibility of having one of my programs go astray and modify the Editor/Assembler once it was in left me in a cold sweat.

Thus I was prompted to write Micromonitor. It's a compromise routine that loads up through my reader in about 50 seconds. The program occupies 159 bytes and is relocatable anywhere without change. I have it located near the top of memory ($2F50_H$–$2FEE_H$) in my 12K system. The program could be used in a system with as little as ½K and be practical. Micromonitor uses memory locations 0000_H through 0003_H as scratchpad for temporary data storage. It is expandable (through the subroutine feature listed below) and does depend on MIKBUG's being present.

Let me list the main features, then I'll show you how to use it with the program example I have included.

Program Features

Memory Load: Allows bytes to be entered directly into memory with error correction and address display upon demand.

Memory Display: Prints contents of memory in lines 16 bytes long with an address at the start of each line. I've found this 16-byte format to be a big assist in computing branches.

Go-To-Subroutine-And-Back: Allows testing of subroutines and provides a method of adding features to Micromonitor.

Go-To-User's-Program: Loads up MIKBUG's program counter (memory locations $A048_H$ and $A049_H$) immediately without going through MIKBUG.

How to Use the Program

Fig. 1 shows a program entry through Micromonitor. The asterisk, L, is where I loaded Micromonitor through the reader using MIKBUG. $A048_H$ and $A049_H$ were then loaded with Micromonitor's starting address, G was typed and Micromonitor prompted with CR, LF and @. I typed the address to be

```
*L
*M A048
*A048 70  2F
*A049 3E  50
*A04A 8C
*G
@0100 L
0100  BD 2F 71 CE 01 40 BD E1 AC 81 0D 27 05 A7 00 08 20 F4
0112  86 20 A7 00 86 4F <
0117  42 A7 01 86 4F A7 02 A7 04 9<86 5A A7 03 86 04 A7 05
0128  BD 2F 71 BD 01 36 CE 01 40 BD E0 7E 20 CA CE FF FF
0139  09 01 01 01 20 <
013D  26 FA 39
0140  @
@0100  013F
0100  BD 2F 71 CE 01 40 BD E1  AC 81 0D 27 05 A7 00 08
0110  20 F4 86 20 A7 00 86 42  A7 01 86 4F A7 02 A7 04
0120  86 5A A7 03 86 04 A7 05  BD 2F 71 BD 01 36 CE 01
0130  40 BD E0 7E 20 CA CE FF  FF 09 01 01 01 26 FA 39
@0136 S
@0100 G
HELLO
HELLO BOZO

*
```

Fig. 1.

Fig. 2.

```
                        NAM    MICR      MON
                 *MICRO MONITOR
                 *ENHANCEMENT TO MIKBUG
                 *PROVIDES FAST LOAD, DISPLAY, GO AND
                 *SUBROUTINE TEST
                 *WRITTEN BY DAVE TIETZ
                        OPT    O,S
        A041     STACK  EQU    $A041
        E0CA     OUT2HS EQU    $E0CA
        E0C8     OUT4HS EQU    $E0C8
        E1D1     OUTEEE EQU    $E1D1
        E1AC     INEEE  EQU    $E1AC
        E10F     MIKGO  EQU    $E10F
        E047     BADDR  EQU    $E047
        A048     PC     EQU    $A048
                 *
2F50                    ORG    $2F50
                 *
                 *INITIALIZE
                 *
2F50 8E A041            LDS    #STACK
2F53 8D 1C       START  BSR    CRLF
2F55 86 40              LDA A  #$40
2F57 8D 2C              BSR    OUTEM     DISPLAY CR, LF, "@"
                 *
                 *CALL ADDRESS AND COMMAND
                 *
2F59 8D 25              BSR    BADDRM    BUILD ADDRESS
2F5B 8D 2B              BSR    INEEM     CALL COMMAND
2F5D 81 4C              CMP A  #$4C      IS IT A "L"?
2F5F 27 50              BEQ    LOAD
2F61 81 20              CMP A  #$20      IS IT A " "?
2F63 27 2C              BEQ    DISP
2F65 81 47              CMP A  #$47      IS IT A "G"?
2F67 27 22              BEQ    GO
2F69 81 53              CMP A  #$53      IS IT A "S"?
2F6B 26 E6              BNE    START     RESTART ON DEFAULT
                 *
                 *SUBROUTINE TEST
                 *
2F6D AD 00              JSR    00,X      JMP TO USER'S SUBROUTINE
2F6F 20 E2              BRA    START     USER SUBROUTINE DONE, RESTART
                 *
                 *SUBROUTINES
                 *
2F71 86 0D       CRLF   LDA A  #$0D
2F73 8D 10              BSR    OUTEM     OUTPUT CARRIGE RETURN
2F75 86 0A              LDA A  #$0A
2F77 8D 0C              BSR    OUTEM     OUTPUT LINE FEED
2F79 8D 00              BSR    NULL2     OUTPUT 4 NULLS
2F7B 4F          NULL2  CLR A
2F7C 8D 07              BSR    OUTEM
2F7E 20 05              BRA    OUTEM     OUTPUT TWO NULLS AND RTS
2F80 BD E047     BADDRM JSR    BADDR     BUILD ADDRESS IN IXR
2F83 86 20       SPOUT  LDA A  #$20
2F85 7E E1D1     OUTEM  JMP    OUTEEE    OUTPUT CHARACTER IN ACC A
2F88 7E E1AC     INEEM  JMP    INEEE     CALL CHARACTER FROM TERMINAL
                 *
                 *GO TO USER PROGRAM
                 *
2F8B FF A048     GO     STX    PC        LOAD MIKBUG PROGRAM COUNTER
2F8E 7E E10F            JMP    MIKGO     JMP TO MIKBUG'S GO
                 *
                 *DISPLAY MEMORY CONTENTS
                 *
2F91 DF 00       DISP   STX    $00       SAVE STARTING ADDR
2F93 8D EB              BSR    BADDRM    GET END ADDR
2F95 08                 INX              ADD 1 FOR TEST
2F96 DF 02              STX    $02       SAVE END ADDR
2F98 DE 00              LDX    $00       RESTORE START ADDR
2F9A 8D 38       NEWLIN BSR    ECHO      ECHO ADDR ON NEW LINE
2F9C C6 11              LDA B  #$11      SET BYTE COUNT TO 17(BASE 10)
2F9E 5A          AGAIN  DEC B
```

loaded, Micromonitor inserted a space and I typed the L. Micromonitor echoed that address on the next line followed by two spaces. Then I typed code in with Micromonitor inserting the spaces between the bytes. After I filled a line, I hit return, and Micromonitor gave the TTY a CR, LF and printed the next address to be loaded (hitting return after any byte entry displays the address of the next byte to be loaded).

Notice that I made some mistakes (for illustration only, of course), and typing < after either a nibble or byte backs up Micromonitor to allow reentry of the data. The < after a byte entry echoes the current memory location to be loaded on the next line; after a nibble, it just continues with the correct byte entry.

When entering the program, I hit carriage return to get the address echo (so I would know program length). After I typed @, which returned me to Micromonitor, I entered the starting address of my program, a space and the ending address. Micromonitor gave me the listing shown in Fig. 1 (free of the <s and teardrops I always have on the paper during program entry).

Notice that the address of the first byte is printed, then eight bytes, and extra space and eight more bytes. If your listings start at $XXX0_H$ memory location (per the example), the extra space is between $XXX7_H$ and $XXX8_H$ and saves some counting to identify bytes. Micromonitor gives a complete listing up to and including the byte at the ending address.

After Micromonitor gave me the listing, it again prompted with @. The program has a time delay subroutine located at memory location 0136_H. I tried it by entering the subroutine's address, followed by an S. The subroutine ran and Micromonitor returned again.

Finally, I ran the complete program. After getting Micromonitor's @, I typed the starting address of the program and then G. This loaded $A048_H$ and $A049_H$ and executed the MIKBUG GO. The program ran, and I hit reset to return to MIKBUG. Now, if I want to get back to Micromonitor, I must use MIKBUG to reload $A048_H$ and $A049_H$ with Micromonitor's starting address and type G.

```
2F9F 27 F9             BEQ    NEWLIN   PRINTED 16 BYTES, DO NEW LINE
2FA1 C1 08             CMP B  #$08     PRINTED 8 BYTES?
2FA3 26 02             BNE    ON       NOPE, KEEP GOING
2FA5 8D DC             BSR    SPOUT    YUP, INSET SPACE
2FA7 BD E0CA ON        JSR    OUT2HS   OUTPUT BYTE TO TERMINAL
2FAA 9C 02             CPX    $02      IXR = END ADDR?
2FAC 26 F0             BNE    AGAIN    NO, DO AGAIN
2FAE 20 A3             BRA    START    DONE, RESTART
                *
                *LOAD MEMORY
                *
2FB0 09         BACKUP DEX             MOVE BACK ONE MEM LOC
2FB1 8D 21      LOAD   BSR    ECHO     ECHO MEM LOC ON NEW LINE
2FB3 8D D3      CALL   BSR    INEEM    CALL 1ST NIBBLE
2FB5 81 3C             CMP A  #$3C     IS IT "<"?
2FB7 27 F7             BEQ    BACKUP   BACK UP ONE LOCATION
2FB9 81 0D             CMP A  #$0D     IS IT "CR"?
2FBB 27 F4             BEQ    LOAD     START NEW LINE
2FBD 81 40             CMP A  #$40     IS IT "@"?
2FBF 27 92             BEQ    START
2FC1 8D 1F             BSR    MKHEX    TEST FOR "<", CONVERT ASCII
2FC3 48                ASL A           MOVE
2FC4 48                ASL A           MASKED ASCII
2FC5 48                ASL A           TO MOST
2FC6 48                ASL A           SIGNIFCANT NIBBLE
2FC7 16                TAB
2FC8 8D BE             BSR    INEEM    CALL 2ND NIBBLE
2FCA 8D 16             BSR    MKHEX    TEST AND CONVERT
2FCC 1B                ABA             ADD TWO NIBBLES
2FCD A7 00             STA A  00,X     LOAD MEM LOC
2FCF 08                INX             POINT TO NEXT
2FD0 8D B1             BSR    SPOUT    INSERT A SPACE
2FD2 20 DF             BRA    CALL     GET NEXT BYTE
2FD4 8D 9B      ECHO   BSR    CRLF
2FD6 DF 00             STX    $00      IXR TO MEMORY
2FD8 CE 0000           LDX    #$0000   POINT TO MEMORY
2FDB BD E0C8           JSR    OUT4HS   DISPLAY IXR CONTENTS
2FDE DE 00             LDX    $00
2FE0 20 A1             BRA    SPOUT    OUTPUT SPACE AND RTS
2FE2 81 3C      MKHEX  CMP A  #$3C     IS IT A "<"?
2FE4 27 CD             BEQ    CALL     YUP, IGNORE 1ST NIBBLE
2FE6 80 30             SUB A  #$30
2FE8 81 09             CMP A  #$09     IS IT NUMBER?
2FEA 2F 02             BLE    OUT
2FEC 80 07             SUB A  #$07     ASSUME IT'S A LETTER
2FEE 39         OUT    RTS
                       END
STACK  A041
OUT2HS E0CA
OUT4HS E0C8
OUTEEE E1D1
INEEE  E1AC
MIKGO  E10F
BADDR  E047
PC     A048
START  2F53
CRLF   2F71
NULL2  2F7B
BADDRM 2F80
SPOUT  2F83
OUTEM  2F85
INEEM  2F88
GO     2F8B
DISP   2F91
NEWLIN 2F9A
AGAIN  2F9E
ON     2FA7
BACKUP 2FB0
LOAD   2FB1
CALL   2FB3
ECHO   2FD4
MKHEX  2FE2
OUT    2FEE

TOTAL ERRORS 00000
```

Concluding Comments

Fig. 2 is a listing of Micromonitor with my comments. The very top bytes of memory have been reserved for any patches I may wish to put in. I chose not to have Micromonitor test for valid hex numbers while loading. If you wish, the MKHEX subroutine could be expanded to test for hex numbers similar to MIKBUG's INHEX subroutine (it takes ten extra bytes).

Because Micromonitor is in MIKBUG's BADDR subroutine after its prompt, typing any non-hex character after the @ will patch you to MIKBUG, and if the program counter isn't changed, typing G after MIKBUG's * will run Micromonitor.

I have used Micromonitor as my main programming monitor for over a year. Even now that I have an Editor/Assembler running on my system, I find numerous occasions to use it. It's still faster than my assembler for memory dumps and quick program patches. ■

DATECK

A program subroutine for date checking and formatting.

J. Tom Badgett
400 Albemarle Street
Bluefield WV 24701

One of my pet peeves is a computer program that requires the operator to remember important details about entering data from the keyboard. This kind of programming is time consuming and wasteful of the very features that make a computer useful in the first place. Moreover, it can lead to improper data entry.

For example, consider the date entered as part of a business program. The software may use this information for aging of accounts, checking records for proper sequence of data entry and the like. Many programs, however, depend on the operator to give the computer a proper date to work with. Some will even accept absolute garbage in the date-entry routine.

Entering Dates

Many programmers, for example, use a single string to enter the date for a given program run. The input statement for this type of date entry might be of the form:

```
10 INPUT "ENTER TODAY'S DATE";D$
```

The operator then enters the date in whatever form he desires: 11/20/78, 11,20,78, or 20 November 1978. He might also enter an invalid date—that is, accidentally enter 32 days in a month. With a single string for input he might even enter complete garbage, either accidentally or maliciously. The computer doesn't care what you type in response to the sample input statement above. It simply is looking for some value for D$.

Here's a short, simple date-entry routine that I now use on all my programs requiring date entry from the keyboard. This routine accepts the date in three separate strings: one for the month, another for the day and a third for the year. Most people seem to expect to enter the date as a series of numbers in the form: 11,20,78. By using three separate entries, you make it easier for the computer to inspect the date you've entered to determine its validity.

I use string entry instead of numerical input because it gives me more versatility in programming. I can make several checks on the strings entered (see Program 1) to cause the program to do a variety of things instead of what it would do normally. Also, you can use this string versatility to format the date input however is convenient. See Program 2 for alternate date-entry scheme uses. You can use this MID$ search routine to look for any type of special character in any string. It is a useful BASIC command to help install program "checks."

```
10 INPUT "PLEASE ENTER TODAY'S DATE ";D$,DI$,Y$
20 IF D$="MENU" THEN 100
30 IF D$="X" THEN CLOSE: END
40 IF D$="HELP" THEN 1000 :  REM   PROGRAM INSTRUCTIONS
50 IF D$="BYE" THEN CLOSE: RUN"CONMND
60 (((Continue with as many similar checks as necessary, then
     go on with the main part of the program))).
```

Program 1.

```
RUN

ENTER TODAY'S DATE (M,D,Y)  ? 11,31,78

DATE INVALID--PLEASE REENTER (M,D,Y): : : : : ? 11,20,78

. . . . . .20/NOVENBER/78. . . . . .
IS DATE OK  ( Y OR NO ):::::: ? Y

?RG ERROR IN  40220
OK

RUN

ENTER TODAY'S DATE (M,D,Y)  ? 2,29,78

IS THIS A LEAP YEAR  (Y OR N)  ? N

DATE INVALID--PLEASE REENTER (M,D,Y): : : : : ? 2,28,78

. . . . . .28/FEBRUARY/78. . . . . .
IS DATE OK  ( Y OR NO ):::::: ? Y

?RG ERROR IN  40220
OK
```

Sample run. The computer requires reentry when invalid dates—November 31 and February 29, 1978—are entered. RG ERROR IN 40220 means the computer didn't know where to go after the RETURN statement.

The Program

Now, refer to Program 3 for the following discussion. The line numbers start at 40000 because I want to use it as a subroutine in many of my programs. That way I can load it into workspace from disk or cassette before I start to write a program and don't have to key it in every time.

Lines 40000-40002 simply help me remember what this particular subroutine does. If you can afford the memory, always document your software in some manner.

Lines 40010-40040 set up a matrix containing the months of the year. The routine uses this information to print out the date for checking in the form: 20/NOVEMBER/78. I prefer this format when the date is to be part of the printed output of a program.

Lines 40050-40060 contain more documentation to help me use the subroutine. After several months without using a particular piece of software I sometimes forget how to use it. In this case, if you forget to dimension the variable M$, your computer will likely return an error because most BASICs limit the size of a matrix to 10 unless you tell the computer otherwise.

Lines 40065-40070 accept the date information from the keyboard. Various checks are performed on the input data in lines 40080-40120. If you enter either the month or the day as a number less than 1, the program will go to line 40800 to print an error message and ask for the information again. Similarly, if the number of days in the month is over 31, the computer will spot the error and require reentry of the data. If the operator attempts to input more than two numbers for the year, the program will perceive this as an error and go to

```
5 DIM D$(17)
10 INPUT "ENTER TODAY'S DATE  (MM/DD/YY) ";D$
12 FL=0
15 A=0
20 FOR J=1 TO 17
30 D$(J)=MID$(D$,J,1)
35 X=X+1
40 IF D$(J)="/" THEN 100
60 NEXT J
100 FOR I=A+1 TO X-1
110 DATE$=DATE$+D$(I)
120 NEXT
125 GOSUB 40000
130 A=X
140 FOR J=X+1 TO 17
145 IF FL=3 THEN END
150 GOTO 30
40000 PRINT "DATE$= ";DATE$
40002 FL=FL+1
40005 PRINT:DATE$=""
40010 RETURN
OK



RUN
ENTER TODAY'S DATE  (MM/DD/YY) ? 11/20/78
DATE$= 11

DATE$= 20

DATE$= 78


OK


RUN
ENTER TODAY'S DATE  (MM/DD/YY) ? 20/NOVEMBER/1978
DATE$= 20

DATE$= NOVEMBER

DATE$= 1978

OK
```

Program 2. Alternate date-entry schemes.

line 40800.

At line 40130 the string for the month is converted to a number for easier manipulation, then several more checks are performed at lines 40140 and 40150. If the 4th, 6th, 9th or 11th months (months with only 30 days) are entered, the subroutine at line 41500 will check to make sure no more than 30 days have been entered for the date in that month. If the date is in February (D = 2) and more than 28 days are entered, the program will go to line 42000 to determine if this is a leap year. If it is not a leap year then the date will be invalid and must be reentered.

Finally, the entered date is printed and the operator is asked to verify it. If it is OK then the program will return to the main program via the RETURN statement in line 40220. If the date needs to be changed at this point this program will go to line 40065 for the date to be reentered.

Note that another simple, but very important, type of data check is made in lines 40220 and 40230. Without line 40230 the program would function OK as long as the operator entered the expected "Y" or "N," "YES" or "NO."

Suppose something else were entered? The program would "fall through" to line 40240 and go to the head of the program for reentry of the date, even if this were not what the operator intended. In this example little harm would be done, but suppose the asked-for input were a complicated formula or important financial information? An accidental entry of the wrong letter would dump the data and require reentry—perhaps compounding the possibility of an error occurring the second time.

Program Use

Program 2 uses the MID$ BASIC command to search a string for certain limiters or identifiers. In this case I've used "/" as the limiter. A loop is set up in line 20 for a maximum of 17 characters (the maximum number likely with a date). The command in line 30 looks at each character in D$, one by one.

In line 40, when the "/" is found, the program leaves the loop and goes to line 100. There another loop is established to build a new string, DATE$. This string consists of all the numbers or letters from the beginning of D$ to, but not including, the first "/."

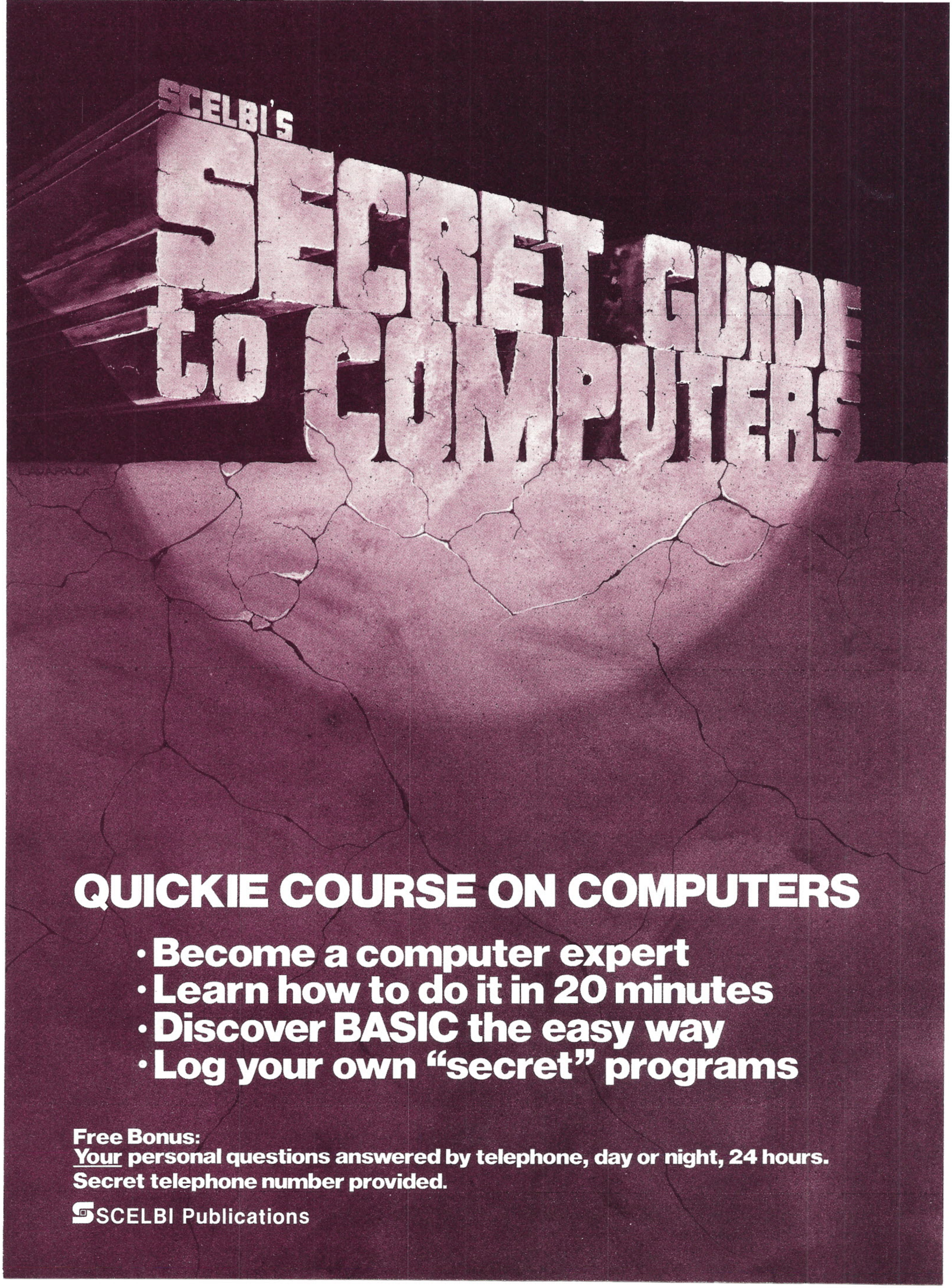
In this example I've included a print statement at line 40000 to show how the string is built, but normally this is where a routine similar to the one in Program 3 resides. After checking the first entry, this program returns to look at the rest of the string. The loop is reset at line 140 to pick up where the last one left off (i.e., after the first "/"), and it searches for the next limiter, and so on. In this way various date-entry routines can be used, checking each segment—day, month and year—for validity. The slight advantage of this arrangement is that the date, including the name of the month in whatever form is convenient, may be entered as long as the limiter "/" is properly placed.

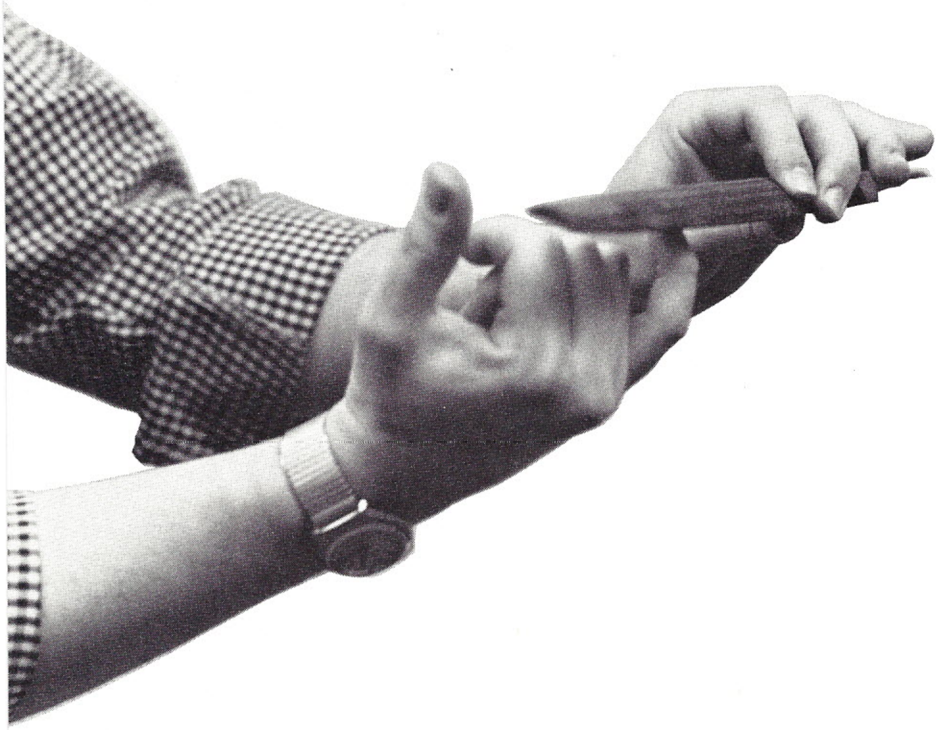
Conclusion

Whenever possible catch all possible situations with software before a serious program or bookkeeping mistake occurs. This little date-checking routine is just one example. Give serious considerations to all programs that require operator input and make sure there are "catches" in your software to handle careless data entry. Such precautions during the programming stage make your computer easier and more reliable to use. ■

```
40000 REM ------------------------------------------------------------
40001 REM          Check/Format Date Subroutine
40002 REM ------------------------------------------------------------
40010 M$(1)="JANUARY":M$(2)="FEBRUARY":N$(3)="MARCH":N$(4)="APRIL"
40020 M$(5)="MAY":M$(6)="JUNE":M$(7)="JULY":M$(8)="AUGUST"
40030 M$(9)="SEPTEMBER":M$(10)="OCTOBER":N$(11)="NOVEMBER"
40040 M$(12)="DECEMBER"
40050 REM ***** Be sure to dimension N$ to 12 at head of Program.
40055 REM ***** Most BASICs automatically limit variables to 10
40060 REM ***** without additional dimensioning.
40065 PRINT: PRINT "ENTER TODAY'S DATE (M,D,Y)  ";
40070 INPUT D$, D1$, Y$
40080 IF LEN(Y$)<>2 THEN GOSUB 40800:GOTO 40070
40090 IF VAL(D$)>12 THEN GOSUB 40800:GOTO 40070
40100 IF VAL (D$)<1 THEN GOSUB 40800: GOTO 40070
40110 IF VAL (D1$)>31 THEN GOSUB 40800:GOTO 40070
40120 IF VAL (D1$)<1 THEN GOSUB 40800:GOTO 40070
40130 D=VAL(D$)
40140 IF D=4 OR D=6 OR D=9 OR D=11 THEN 41500
40150 IF D=2 AND VAL(D1$)>28 THEN 42000
40160 DATE$=D1$ + "/" + M$(D) + "/" + Y$
40200 PRINT:PRINT ". . . . . .";DATE$;". . . . . ."
40210 INPUT "IS DATE OK  ( Y OR NO ):::::  ";Z$
40220 IF LEFT$(Z$,1)="Y" THEN RETURN
40230 IF LEFT$(Z$,1)<>"N" THEN 40200
40240 GOTO 40065
40800 PRINT:PRINT "DATE INVALID--PLEASE REENTER (M,D,Y): : : : : ";
40820 RETURN
41500 IF VAL(D1$)<31 THEN 40150
41510 GOSUB 40800:GOTO 40070
42000 PRINT: INPUT "IS THIS A LEAP YEAR  (Y OR N)  ";L$
42010 IF LEFT$(L$,1)="Y" THEN 40160
42020 IF LEFT$(L$,1)<>"N" THEN 42000
42030 GOSUB 40800: GOTO 40070
OK
```

Program 3.


SCELBI'S
SECRET GUIDE to COMPUTERS
QUICKIE COURSE ON COMPUTERS
• Become a computer expert
• Learn how to do it in 20 minutes
• Discover BASIC the easy way
• Log your own "secret" programs
Free Bonus:
Your personal questions answered by telephone, day or night, 24 hours.
Secret telephone number provided.
SCELBI Publications



FILEIT

Last month, we had, as you might say, a "screening" of this program.

Forest E. Myers
5114 Garnett St.
Shawnee KS 66203

If you have had a chance to implement my SCREEN program in last month's issue, you are probably waiting to implement its complement program, FILEIT. As you will recall, the purpose of SCREEN was to create a data entry screen to be used by FILEIT, a program to create and manipulate a data base. Although the task of bringing FILEIT up on your system will be somewhat more difficult than with SCREEN, the task is not insurmountable, and the rewards for doing so are considerable.

FILEIT is a lengthy program. For some systems it may be too large to run as a single program. This is not a serious problem since almost every option in FILEIT is done via subroutine calls. As a result, FILEIT can be thought of as groupings of modules, each performing a separate task.

To run FILEIT on a smaller system, therefore, requires only that the necessary modules to perform a given task be made into a separate program. The most logical starting point for this breakout process is to make the data retrieval option (option 3) a separate program.

Table 1. Primary FILEIT variables.

A$	Input/output string
B$	File name string
C$	Utility string for user responses and CONVERT function
D$	Holds user input data
E$	Label name string
F$	Null string. Used to null other strings when necessary
G$	User input string for selection criterion
T$	Used in purge blank record routine to hold nonblank information to fill blank record
B()	Holds record number of blank records in data base
C()	Used to hold column address for cursor control
D()	Length of individual data fields
L()	Length of individual labels
R()	Used to hold row address for cursor control
L8()	Holds variable number in user specified report option
N1()	Holds numeric field designators
R8()	Holds and row and position address for user specified reports
B1	Holds the number of records included in a 256 byte block
D1	Used to determine the beginning position of a given data field within a record
I3	Holds the variable number selected by the user to serve as as the selection criterion for a record's inclusion in an output report.
J1	Beginning address of a given label in the Label String E$
K1	Ending address of a given label in the Label String E$
L1	1st CRT line data entry screen is to be displayed
P8	Holds variables relative print position on a given row in user specified report option
P9	Holds line length of printer or CRT device in user specified report option
R0	User specified record number to be deleted
S1	Switch used to tell screen print routine that screen is to be used for output and not input.
S2	Switch = 1 if printer is on.
S3	Switch =1 in retrieve option to prevent printing of record number on selection critierion screen
S7	Switch = 1 if correct record or delete record options being used. Causes screen print routine to output a single record.
S8	Switch = 1 when creation of data base option first entered. Allows the creation of a single data base during a given run of FILEIT.
S9	Switch = 1 keeps user from having to enter I/O device number and file name with each new option used.
Z3	Holds number of records entered by user
D	Holds actual length of data record
E	Error code conditions for I/O operation to disk drives
F	Disk drive number where data base is to stored or is stored
I	Block counter in the creation option
O	Option number selected by user from option list
R	Maximum number of records in data base lists
Z	Total number of labels in a record

Similarly, the compress file option (option 6) could be made into a separate program.

FILEIT provides the user with several basic options that allow creation of a new data base or modification of an existing data base. These options are:

1. Create data base
2. Add records to data base
3. Retrieve records from data base
4. Correct records
5. Delete records
6. Compress data base

The discussion to follow will center on using these options. Minimal attention will be given to the source listing for FILEIT except to mention that numerous REMark statements have been added to the program listing to aid the deciphering process. Additionally, some of the more commonly used variables in the program are shown in Table 1. I hope these two aids will shed some light on what is going on in the program.

Option 1

Create data base. This option is used to create a new data base. As part of the creation process, FILEIT utilizes a previously created data entry screen file output by the SCREEN program. The user supplies the screen file name and the storage device number on which it is located.

The user then supplies the maximum number of records that could conceivably be included in the data base. Since the user's response is used to allocate diskette space to the data base, care should be used to ensure that sufficient space is allocated for any future additions to the data base.

After space is allocated, information in the data entry screen file is transferred into records 0 through 3 of the data base, and a data entry screen for the first data record is presented to the user. From here the user enters information in response to the prompting "?" after each label.

After information is supplied for each label, an opportunity is given for correcting information entered. To correct an erroneous entry, simply type in its label. FILEIT will respond by blanking out the previously entered information with "c" indicating the length of the data field. After this, enter the correct information.

If no corrections are to be made, press RETURN. FILEIT will respond by asking if more records are to be entered. If so, pressing RETURN or entering a "Y" will cause the data entry screen to be refreshed for entry of a new record. If an "N" is entered, FILEIT will store any unstored information entered, update record zero for number of records entered and return the user to the options menu.

Option 2

Add records to data base.

FILEIT program.

```
10  CLOSE (0,E) : CLOSE (1,E)
20  DIM A$(256),B$(10),C$(10),D$(256),E$(512),C(30),D(30)
30  DIM R(30),L(30),F$(256)
40  DIM G$(256),T$(256),B(30)
50  DIM R8(30),L8(30),N1(30)
60  #"" : FOR I=1 TO 7 : #" " : NEXT I
70  #"                        FILE CREATION PROGRAM"
80  A$=" " : FOR I=1 TO 8 : A$=A$+A$ : NEXT I
90  D$=A$ : B$=A$(1,10) : C$=B$ : E$=A$+D$ : F$=A$
100  G$=A$ : T$=A$
110  FOR I=1 TO 5 : #" " : NEXT I
120  INPUT "Enter input/output device number (0 or 1) ",F
130  IF F<0 OR F>1 THEN #"Only 0 or 1 allowed " : GOTO 120
140  #""
150  #"OPTIONS"
160  #"    1. Create file"
170  #"    2. Add records to file "
180  #"    3. Retrieve records from file"
190  #"    4. Correct records "
200  #"    5. Delete records "
210  #"    6. Compress file "
220  #"    7. End processing"
230  #" "
240  INPUT "Enter option number ",O
250  IF O<1 OR O>7 THEN 140
260  IF O=1 THEN 690
270  IF O=7 THEN 7300
280  IF S9=1 THEN 630
290  INPUT "Enter name of data file ",B$
300  IF LEN(B$)<>6 THEN #"6 characters must be entered " : GOTO 290
310  S9=1
320  B$=B$+".DA"+CHR$(0)
330  OPEN (1,E,B$,3,F)
340  IF E>=1 THEN 7160
350  GET (1,E,A$,0)
360  IF E>1=1 THEN 7160
370  C$=A$(241,242) : CONVERT C$ TO Z : REM Z=NUMBER OF LABELS
380  C$=A$(243,244) : CONVERT C$ TO L1 : REM L1=1ST SCREEN LINE
390  C$=A$(245,246) : CONVERT C$ TO B1 : REM B1=RECORDS PER BLOCK
400  C$=A$(247,249) : CONVERT C$ TO D : REM D=LENGTH OF RECORD
410  C$=A$(250,252) : CONVERT C$ TO Z3 : REM Z3=NUMBER OF RECORDS
420  C$=A$(253,255) : CONVERT C$ TO R : REM R=MAX NUMBER OF RECORDS
430  FOR I=1 TO Z
440    C$=A$(I*2-1,I*2) : CONVERT C$ TO L(I)
450    C$=A$(119+I*2,120+I*2) : CONVERT C$ TO D(I)
460    NEXT I
470  FOR I=1 TO 2
480    GET (1,E,A$,I)
490    IF E>0 THEN 7160
500    E$(I*256-255,I*256)=A$
510    NEXT I
520  GET (1,E,A$,3)
530  IF E>0 THEN 7160
540  FOR I=1 TO Z
550    C$=A$(I*2-1,I*2) : CONVERT C$ TO R(I)
560    C$=A$(119+I*2,120+I*2) : CONVERT C$ TO C(I)
570    NEXT I
580  GET (1,E,A$,4)
590  FOR I=1 TO Z
600    C$=A$(I,I)
610    CONVERT C$ TO N1(I)
620    NEXT I
630  IF O=2 THEN 4070
640  IF O=3 THEN 4480
650  IF O=4 THEN 4240
660  IF O=5 THEN 5710
670  IF O=6 THEN 6680
680  REM***************  CREATE DATA BASE OPTION  ***************
690  IF S8=1 THEN #"This option not allowed "
700  IF S8=1 THEN FOR I=1 TO 500 : # : NEXT I
710  IF S8=1 THEN 140
720  S8=1
730  INPUT "Enter screen file name ",B$
740  IF LEN(B$)<>6 THEN #"6 characters must be entered " : GOTO 730
750  B$=B$+".SE"+CHR$(0)
760  OPEN (0,E,B$,2,F)
770  IF E>=1 THEN 7160
780  INPUT "Enter name for data file ",B$
790  IF LEN(B$)<>6 THEN #"6 characters must be entered " : GOTO 780
800  INPUT "Maximum number of records to be in file ",R
810  B$=B$+".DA"+CHR$(0)
820  IF E>1 THEN 7160
830  GET (0,E,A$,0)
840  IF E>=1 THEN 7160
850  C$=A$(241,242) : CONVERT C$ TO Z : REM NUMBER OF LABELS
860  C$=A$(243,244) : CONVERT C$ TO L1 : REM 1ST SCRREN LINE
870  C$=A$(245,246) : CONVERT C$ TO B1 : REM BLOCKING FACTOR
880  C$=A$(247,249) : CONVERT C$ TO D : REM RECORD LENGTH
890  R2=INT(R/B1)
900  A=R-B1*R2
910  R2=R2+5
920  IF A>0 THEN R2=R2+1
930  OPEN (1,E,B$,3.F,R2)
940  IF E>1 THEN 7160
950  A$=F$ : REM BLANK OUT I/O STRING
960  #"" : #"Making room for file on diskette "
970  FOR I=0 TO R2 : PUT (1,E,F$,I) : NEXT I
980  REM GET INFORMATION FROM DATA ENTRY SCREEN FILE
990  REM AND PUT IT IN RECORDS 0-4 OF DATA BASE FILE.
1000  GET (0,E,A$,0)
1010  IF E>=1 THEN 7160
1020  PUT (1,E,A$,0)
1030  IF E>=1 THEN 7160
1040  FOR I=1 TO Z
1050    C$=A$(I*2-1,I*2) : CONVERT C$ TO L(I) : REM LABEL LENGTHS
1060    C$=A$(119+I*2,120+I*2) : CONVERT C$ TO D(I) : REM DATA FIELD LENGTH
1070    NEXT I
1080  FOR I=1 TO 2
1090    GET (0,E,A$,I)
1100    IF E>=1 THEN 7160
1110    PUT (1,E,A$,I)
1120    IF E>=1 THEN 7160
1130    E$(I*256-255,I*256)=A$
1140    NEXT I
1150  GET (0,E,A$,3)
1160  IF E>=1 THEN 7160
1170  PUT (1,E,A$,3)
1180  IF E>=1 THEN 7160
1190  FOR I=1 TO Z
1200    C$=A$(I*2-1,I*2) : CONVERT C$ TO R(I) : REM ROW ADDRESSES
1210    C$=A$(119+I*2,120+I*2) : CONVERT C$ TO C(I) : REM COL ADDRESSES
1220    NEXT I
1230  GET (0,E,A$,4)
1240  FOR I=1 TO Z
1250    C$=A$(I,I)
1260    CONVERT C$ TO N1(I)
1270    NEXT I
1280  REM CLOSE DATA ENTRY SCREEN FILE SINCE ITS DATA IS
1290  PUT (1,E,A$,4)
1300  REM NOW IN THE FIRST RECORDS OF DATA BASE FILE
1310  CLOSE (0,E)
1320  IF E>0 THEN 7160
1330  GOSUB 1350 : REM CALL DATA ENTRY SCREEN ROUTINE
1340  GOTO 140
1350  REM ***********  DATA ENTRY SCREEN ROTUINE ***************"
1360  REM I IS BLOCK COUNTER.  IT IS SET TO 5 FOR FILE CREATION.
1370  REM RECORDS 0-4 HOLD HOUSEKEEPING INFORMATION FOR DATA BASE.
1380  REM      Z3 HOLDS ACTUAL NUMBER OF RECORDS ENTERED
1390  REM      B2 SET TO 1 FOR FILE CREATION.  CORRECTION OPTION
1400  REM         SETS B2 TO RECORD SELECTED BY USER TO BE CORRECTED
1410  REM       J HOLDS CURRENT RECORD WITHIN BLOCK
1420  REM      S1 SWITCH THAT TELLS SCREEN PRINT ROUTINE THAT SCREEN
1430  REM         IS TO BE USED FOR OUTPUT RATHER THAN INPUT
1440  REM       L POINTS TO CURRENT LABEL FOR WHICH DATA IS TO BE
1450  REM         ENTERED
1460  REM       Z HOLDS THE NUMBER OF LABELS THAT MAKE UP A RECORD
1470  REM      D$ HOLDS USER INPUT DATA
1480  REM      D1 POINTS TO RECORD POSITION THAT DATA IS TO BE PLACED
1490  REM      Z1 HOLDS ACTUAL AMOUNT OF DATA FIELD SPACE USED
1500  I=5 : Z3=0 : B2=1 : A$=F$
1510  FOR J=B2 TO B1
1520    S1=0
1530    GOSUB 2310 : REM GOTO DATA ENTRY SCREEN PRINT ROUTINE
```

```
1540  FOR L=1 TO Z
1550    CURSOR 15,0
1560    #"                                                            ";
1570    IF Z3=R+1 THEN #"Last record that can be input "; : GOTO 1600
1580    IF N1(L)=1 THEN CURSOR 15,0 : #"Record ";Z3+1;" ** Numeric field";
1590    IF N1(L)=0 THEN CURSOR 15,0 : #"Record ";Z3+1;
1600    CURSOR R(L),C(L)+L(L)
1610    D$=F$
1620    D1=D1+D(L)
1630    INPUT1 D$
1640    Z1=D(L)-LEN(D$)
1650    REM BLANK OUT "x's" THAT WERE NOT COVERED WITH USER INPUT
1660    REM DATA
1670    FOR Z2=1 TO Z1
1680      #" ";
1690      NEXT Z2
1700    #" ";
1710    REM       NOTE:  PROMPTING ? GENERATED BY INPUT1 STATEMENT
1720    REM              IS ERASED BY THE NEXT LINE OF CODE
1730    CURSOR R(L),C(L)+L(L) : #" ";
1740    IF N1(L)=0 THEN 1900
1750    IF LEN(D$)=D(L) THEN T$(1,D(L))=D$ : GOTO 1840
1760    IF LEN(D$)=0 AND N1(L)=1 THEN D$="0"
1770    V1=D(L)-LEN(D$)+1 : V2=V1-1
1780    IF V1<=0 THEN S0=1 : GOSUB 7350 : GOTO 1630
1790    T$(V1,D(L))=D$
1800    FOR K9=1 TO V2
1810      T$(K9,K9)="0"
1820      NEXT K9
1830    D$=T$(1,D(L))
1840    FOR K9=1 TO D(L)
1850      C$=T$(K9,K9)
1860      IF ASC(C$)=46 OR ASC(C$)=45 THEN 1890
1870      IF ASC(C$)<48 OR ASC(C$)>57 THEN GOSUB 7350
1880      IF ASC(C$)<48 OR ASC(C$)>57 THEN EXIT 1630
1890      NEXT K9
1900    A$(J*D-D+(D1-(D(L)-1)),J*D-D+D1)=D$
1910    NEXT L
1920  CURSOR 15,0
1930  #"                                                            ";
1940  CURSOR 15,0
1950  #" ";
1960  CURSOR 15,0
1970  #"Correction ";
1980  INPUT1 C$
1990  REM IF USER WANTS TO CORRECT ERROR GOTO CORRECTION ROUTINE
2000  IF C$="Y" OR C$="y" THEN GOSUB 2790 : REM CALL CORRECT DATA
2010  CURSOR 15,0
2020  #"                                                  ";
2030  CURSOR 15,0
2040  Z3=Z3+1
2050  #"More to be entered ";
2060  INPUT1 C$
2070  IF C$="N" OR C$="n" THEN 2160
2080  NEXT J
2090 B2=1
2100 REM DATA BLOCK MUST BE FULL, WRITE IT OUT TO STORAGE DEVICE
2110 PUT (1,E,A$,I)
2120 IF E>=1 THEN 7160
2130 I=I+1 : A$=F$
2140 GOTO 1510
2150 REM STORE AWAY ANY DATA ENTERED SINCE LAST STORE
2160 PUT (1,E,A$,I)
2170 REM UPDATE RECORD ZERO FOR NUMBER OF RECORDS ENTERED AND
2180 REM MAXIMUM NUMBER OF RECORDS THAT CAN BE PLACED IN FILE
2190 GET (1,E,A$,0)
2200 IF E>=1 THEN 2800
2210 CONVERT Z3 TO C$(###)
2220 A$(250,252)=C$
2230 CONVERT R TO C$(###)
2240 A$(253,255)=C$
2250 REM STORE UPDATED RECORD ZERO
2260 PUT (1,E,A$,0)
2270 IF E>=1 THEN 7160
2280 IF S8=1 THEN CLOSE (1,E) : OPEN (1,E,B$,3,F) : S9=1
2290 RETURN
2300 REM
2310 REM ***********************  SCREEN PRINT ROUTINE  ************"
2320 REM
2330 REM     D1 HOLDS POSITION COUNTER FOR DATA WITHIN RECORD
2340 REM     J1 HOLDS BEGINNING ADDRESS OF LABEL IN E$
2350 REM     S3 SWITCH SET IN RETRIEVE RECORD OPTION
2360 REM     S2 SWITCH SET TO 1 IF PRINTER ON
2370 REM     S1 SWITCH SET TO 1 IF SCREEN FOR OUTPUT NOT INPUT
2380 REM     S7 SWITCH SET TO 1 IN CORRECT OR DELETE RECORDS OPTIONS
2390 REM        TELLS SCREEN PRINT TO OUTPUT A SINGLE RECORD
2400 REM     W9 NUMBER OF LINES TO SEPERATE PRINTED RECORDS
2410 REM     K1 HOLDS ENDING ADDRESS OF LABEL IN E$
2420 #""
2430 D1=0
2440 J1=1
2450 IF S3=1 THEN 2470
2460 IF S2=1 THEN #TAB(60);"Record ";Z4
2470 S3=0
2480 FOR I1=1 TO Z
2490   REM          IF PRINTER ON AND ROW NUMBERS DIFFER, THEN PRINT
2500   REM          A BLANK TO FORCE A CARRIAGE RETURN AND LINE FEED
2510   IF S2=1 AND R(I1-1)<>R(I1) THEN #" "
2520   CURSOR R(I1),C(I1)
2530   K1=J1+L(I1)-1
2540   #E$(J1,K1);
2550   #" ";
2560   J1=K1+1
2570   D1=D1+D(I1)
2580   IF S1=1 THEN 2640
2590   REM DISPLAY "x's" TO INDICATE LENGTH OF DATA FIELD
2600   FOR M1=1 TO D(I1)
2610     #"x";
2620     NEXT M1
2630   GOTO 2660
2640   #A$(J*D-D+(D1-(D(I1)-1)),J*D-D+D1);
2650   IF S2=1 THEN #"  ";
2660   NEXT I1
2670 IF S7=1 THEN RETURN
2680 IF S1=0 THEN 2760
2690 IF S2=1 THEN 2760
2700 CURSOR 15,0
2710 #"                                                            ";
2720 CURSOR 15,0 : IF S2=1 THEN 2760
2730 #"   Record  ";%3I;Z4;"   ";
2740 INPUT1 "Hit CR for next record ",C$
```

This option is used to add records to an existing data base. Records added are appended to those already in the file. This is done automatically and does not require any effort on the user's part.

In the record addition process, FILEIT informs the user of the number of records in the file and the number of records the file can hold. After this, addition of records to the data base proceeds in a manner similar to that described in option 1 when the file was originally created.

Option 3

Retrieve records from data base. The user can have records meeting a specified criterion or all records in the data base output to the CRT or to the system's printer (answer "Y" to the printer request). The user can use the screen entry format as the format for the output, or the user can specify his own format.

To specify his own format, the user inputs a 2 to the suboption request. The user is then asked if a previously specified report format is to be used. If the answer is "Y," the name of the report format file and the disk drive where it is stored is requested. If a carriage return is entered, the user must enter the report format.

To specify a format the user is asked to provide the report line number where he wants a particular label and its associated data field to appear on the report. Additionally, the user is asked to provide the label's relative position on the given line. If a line number of 15 or more is given, the item will not be printed or displayed. Also, report line numbers need not be entered in any particular order since after the entry of each label's line number all label positions are sorted in order.

If too much print information is indicated for a line, a "line too long" message will be displayed. If this occurs, simply put the label on another report line.

One other aspect of option 3 is that the user can specify a single criterion for records' inclusion in a report. Just type in the criterion after the label on the screen. A carriage return will move you through the labels until you reach the one which is wanted for selection purposes.

If all records in the data base are to be included in the report, keep hitting the carriage return until you have passed through all labels. Once a carriage return has been entered for the last label, the report generation process begins. (NOTE: If the user specifies a criterion, the report generation process will begin after the criterion's specification.)

Option 4

Correct records. This option allows the user to correct erroneous records in previously created data files. To accomplish this, the user enters the number of the record to be corrected. FILEIT responds by displaying the record on the CRT and asking for the erroneous data field's identifying label. The correction process then proceeds in a manner identical to that described in correcting records at the time of data file creation.

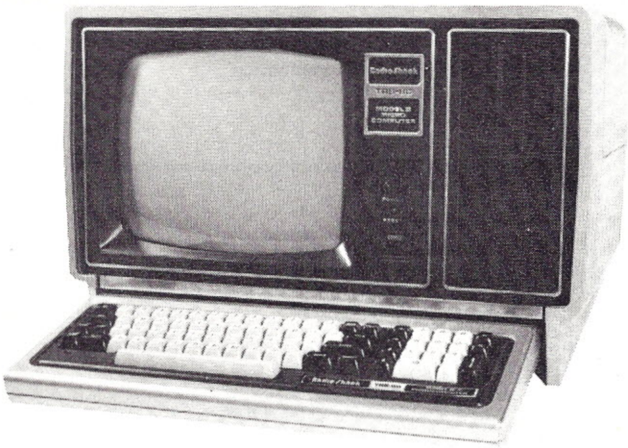
Option 5

Delete records. The delete record option is used to eliminate records from the data base found no longer valid or useful. Records are deleted by blanking them out. Under the retrieve option, these records will appear as blank data. To delete records, the user enters the number of the record to be deleted. FILEIT displays the record and asks if it is to be deleted. Answering "Y" to the query will cause the record to be blanked out.

Option 6

Compress data base. The compress option squeezes blank records out of the data base resulting from record deletions. Calling this option automatically causes FILEIT to move nonblank records from the end of the file into blank spaces in earlier parts of the file.

These two programs, SCREEN and FILEIT, together offer a method of avoiding a lot of the drudgery associated with data files. I hope that they will be a worthwhile addition to your software library. ■



MODEL II 64K TRS-80™

$3500.00

ACS SLASHES PRICES

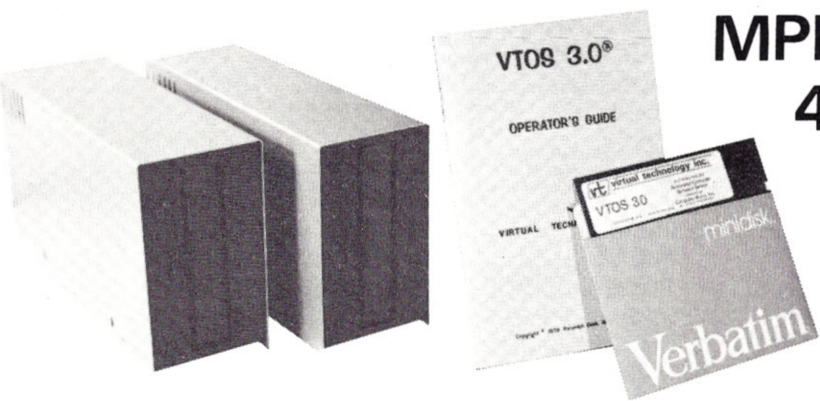
TRS-80™ MODEL I Level II 16K

$699.95

with Keypad
(not shown)

Radio Shack®
EXPANSION INTERFACE

$249.95



MPI B-51 40 TRACK DRIVES

Includes: Power Supply, Case,
4-Drive Cable, and 3.0 DOS
12 ms stepping rate.

$299.95 Complete Package

Master Charge – COD's WELCOME – Visa
CHECKS – 14 days to clear.

Call our Model II Based Computer Bulletin Board System – On line 24 hours a day, 7 days a week – (615) 254-9193

AUTOMATED COMPUTER SOFTWARE SERVICE ✓232

1100 8th Avenue South – Nashville, TN 37203
(615) 244-7021

1-800-251-3493

– Telephone Orders Only –

Send Check or Money Order payable to –
Automated Computer Software Service
1100 8th Avenue, South – Nashville, TN 37203

Quan.	Description	Unit Price	Total

HANDLING CHARGE $1.50
TENN. RES. ADD 6% SALES TAX **TOTAL**

☐ Check
☐ Money Order
☐ MasterCharge
☐ Visa
Card No. ________
Exp. Date ________

Name ________
Address ________
City ________ State ________ Zip ________

TM–TRS-80 is a Trademark of Tandy Corp.

```
2750 #" ";
2760 IF S2=1 THEN FOR W0=1 TO W9 : #" " : NEXT W0
2770 D1=0
2780 RETURN
2790 REM **********  CORRECT DATA WITHIN RECORD ROUTINE  **********"
2800 CURSOR 15,0
2810 #"                                                            ";
2820 CURSOR 15,0
2830 #"Enter label of data field to be corrected ";
2840 INPUT1 D$
2850 J1=1 : D1=0
2860 REM FIND THE LABEL THE USER SAID HAD INCORRECT DATA
2870 FOR I1=1 TO Z
2880   K1=J1+L(I1)-1
2890   D1=D1+D(I1)
2900   IF D$=E$(J1,K1) THEN 3010
2910   J1=K1+1
2920   NEXT I1
2930 CURSOR 15,0
2940 #"                                                            ";
2950 CURSOR 15,0
2960 #"Label not found ";
2970 FOR I=1 TO 500 : # : NEXT I
2980 CURSOR 15,0
2990 #"                                                       ";
3000 GOTO 3350
3010 CURSOR 15,0
3020 #"                                                           ";
3030 CURSOR 15,0
3040 L=I1
3050 IF N1(I1)=1 THEN #"Numeric field ";
3060 REM IF LABEL IS FOUND POSITION CURSOR AND DISPLAY PROMPTING
3070 REM "c's" TO INDICATE LENGTH OF DATA FIELD.
3080 CURSOR R(I1),C(I1)+L(I1)+1
3090 FOR I2=1 TO D(I1) : #"c"; : NEXT I2 : #"  ";
3100 CURSOR R(I1),C(I1)+L(I1)
3110 U8=1 : REM KEEP NUMERIC INFORMATION ROUTINE FROM PRINTING "X"
3120 INPUT1 D$
3130 Z1=D(I1)-LEN(D$)
3140 REM BLANK OUT ANY UNUSED DATA SPACE TO PRETTY UP DISLAY.
3150 FOR J2=1 TO Z1 : #" "; : NEXT J2
3160 #" ";
3170 CURSOR R(I1),C(I1)+L(I1) : #" ";
3180 IF N1(I1)=0 THEN 3330
3190 IF LEN(D$)=D(I1) THEN V1=1 : GOTO 3230
3200 IF LEN(D$)=0 AND N1(I1)=1 THEN D$="0"
3210 V1=D(I1)-LEN(D$)+1
3220 IF V1<=0 THEN S0=1 : GOSUB 7350 : GOTO 3010
3230 T$(V1,D(I1))=D$ : V2=V1-1
3240 FOR K9=1 TO V2
3250   T$(K9,K9)="0"
3260   NEXT K9
3270 D$=T$(1,D(I1))
3280 FOR K9=1 TO D(I1)
3290   C$=T$(K9,K9)
3300   IF ASC(C$)=46 OR ASC(C$)=45 THEN 3320
3310   IF ASC(C$)<48 OR ASC(C$)>57 THEN GOSUB 7350 : EXIT 3010
3320   NEXT K9
3330 A$(J*D-D+(D1-(D(I1)-1)),J*D-D+D1)=F$(J*D-D+(D1-(D(I1)-1)),J*D-D+D1)
3340 A$(J*D-D+(D1-(D(I1)-1)),J*D-D+D1)=D$
3350 S1=1
3360 CURSOR 15,0
3370 #"                                                           ";
3380 CURSOR 15,0
3390 #"More corrections ";
3400 INPUT1 C$
3410 IF C$="Y" OR C$="y" THEN 2790
3420 RETURN
3430 REM
3440 REM
3450 REM **************   SELECTION ROUTINE  ****************
3460 REM
3470 J1=1 : D1=0
3480 #""
3490 GOSUB 2310 : REM CALL SCREEN PRINT ROUTINE
3500 CURSOR 15,0
3510 #"                                                       ";
3520 CURSOR 15,0
3530 #"Enter selection criterion at ? above ";
3540 D1=0 : I3=0 : REM I3 HOLDS NUMBER OF LABEL SELECTED
3550 FOR I1=1 TO Z
3560   CURSOR R(I1),C(I1)+L(I1)
3570   G$=F$
3580   D1=D1+D(I1)
3590   INPUT1 G$(D1-(D(I1)-1),D1)
3600   #" ";
3610   D$=G$(D1-(D(I1)-1),D1)
3620   FOR Z2=1 TO D(I1)
3630     IF G$(D1-D(I1)+Z2,D1-D(I1)+Z2+1)="  " THEN EXIT 3660
3640     NEXT Z2
3650   REM BLANK OUT "x's" IF FIELD IS NOT SELECTED
3660   Z2=D(I1)-Z2 : FOR Z5=1 TO Z2 : #" "; : NEXT Z5
3670   CURSOR R(I1),C(I1)+L(I1) : #" ";
3680   IF G$(D1-(D(I1)-1),D1-(D(I1)-1))=" " THEN 3700
3690   EXIT 3790
3700   NEXT I1
3710 CURSOR 15,0
3720 #"                                                         ";
3730 CURSOR 15,0
3740 #"No selection criterion selected";
3750 FOR C0=1 TO 500 : # : NEXT C0
3760 D$="ALL"
3770 RETURN
3780 REM STORE LABEL SELCECTED BY USER IN I3
3790 I3=I1 : D2=D1
3800 RETURN
3810 REM ********************  READ FILE ROUTINE  *****************
3820 S1=1
3830 B2=INT(B0/B1) : A=B0-B1*B2 : B2=B2+4 : IF A>0 THEN B2=B2+1
3840 IF A=0 THEN A=B1
3850 E2=INT(E0/B1) : E3=E0-B1*E2 : E2=E2+4 : IF E3>0 THEN E2=E2+1
3860 Z4=B0-1
3870 FOR B=B2 TO E2
3880   GET (1,E,A$,B)
3890   IF E>1 THEN #" GET ERROR " : STOP
3900   FOR J=A TO B1
3910     Z4=Z4+1 : IF Z4=E0+1 THEN EXIT 4050
3920     D1=0
3930     FOR L1=1 TO Z
3940       IF D$="ALL" OR D$="all" THEN EXIT 4010
3950       D1=D1+D(L1)
3960       X=J*D-D+(D1-(D(L1)-1))
3970       Y=J*D-D+D1
3980       IF G$(D2-(D(I3)-1),D2)=A$(X,Y) THEN EXIT 4010
3990       NEXT L1
4000     GOTO 4020
4010     GOSUB 2430 : REM CALL SCREEN PRINT ROUTINE
4020     NEXT J
4030   A=1
4040   NEXT B
4050 S1=0 : D1=0
4060 RETURN
4070 REM *******************  ADD RECORDS OPTION  ******************
4080 #""
4090 #"Currently ";Z3;" records in file "
4100 R2=R
4110 #"File can hold as many as ";R2;" records"
4120 #" "
4130 INPUT "Hit CR to begin entry of new data ",C$
4140 X1=INT(Z3/B1)
4150 A=Z3-X1*B1
4160 IF A>0 THEN X1=X1+1
4170 I=4+X1
4180 B2=A+1
4190 IF A=0 THEN I=I+1
4200 GET (1,E,A$,I)
4210 IF E>=1 THEN 7160
4220 GOSUB 1510 : REM CALL DATA ENTRY SCREEN ROUTINE
4230 GOTO 140
4240 REM***************  CORRECT RECORD OPTION  *******************
4250 S7=1 : S1=1 : L8=0
4260 #""
4270 INPUT "Enter record number to be corrected ",L9
4280 L8=INT(L9/B1)
4290 A=L9-L8*B1
4300 IF A>0 THEN L8=L8+1
4310 I=4+L8
4320 L8=L8+4
4330 IF A=0 THEN A=B1
4340 GET (1,E,A$,L8)
4350 IF E>=1 THEN 7160
4360 J=A
4370 GOSUB 2310 : REM CALL SCREEN PRINT ROUTINE
4380 GOSUB 2790 : REM CALL CORRECT DATA ROUTINE
4390 PUT (1,E,A$,L8)
4400 IF E>=1 THEN 7160
4410 CURSOR 15,0
4420 #"                                                          ";
4430 CURSOR 15,0
4440 INPUT1 "Another record to be corrected ",C$
4450 IF C$="Y" OR C$="y" THEN 4260
4460 S1=0 : A$=F$ : S7=0
4470 GOTO 140
4480 REM **************  RETRIEVE RECORDS OPTION  ***************
4490 S1=0
4500 #""
4510 INPUT "OPTIONS  1. Screen report  2. Specified report ",O
4520 IF O<1 OR O>2 THEN 4510
4530 INPUT "Printer on ? (Y/N) ",C$
4540 REM IF USER WANTS PRINTER ON, SET S2=1
4550 IF C$="Y" OR C$="y" THEN S2=1 ELSE S2=0
4560 IF S2=1 THEN INPUT "How many lines between records ",W9
4570 S3=1
4580 INPUT "Enter beginning record to be output ",B0
4590 IF B0<1 THEN #"Record must be one or more " : GOTO 4580
4600 IF B0>Z3 THEN #"Only";Z3;" records in file " : GOTO 4580
4610 #Z3;" records in file"
4620 INPUT "Enter ending record to be output ",E0
4630 IF E0>Z3 THEN #"Only";Z3;" records in file " : GOTO 4620
4640 IF E0<B0 THEN #"Beginning more than ending " : GOTO 4620
4650 INPUT "Want to specify a selection criterion (Y/N ) ",C$
4660 IF C$="Y" OR C$="y" THEN 4710
4670 #""
4680 D$="ALL" : S3=0 : GOTO 4720
4690 REM PROVIDE USER WITH DATA ENTRY SCREEN TO SEE IF ONLY
4700 REM DATA MEETING A SPECIFIED CRITERION IS TO BE OUTPUT.
4710 GOSUB 3450 : REM  CALL SELECTION ROUTINE
4720 REM
4730 IF O=2 THEN 4820
4740 IF S2=1 THEN #"" : #"Output is being printed"
4750 IF S2=1 THEN CLOSE (CRT,E)
4760 IF S2=1 THEN OPEN (PRINTER,E)
4770 GOSUB 3810 : REM  CALL READ DATA BASE ROUTINE
4780 CLOSE (PRINTER,E)
4790 OPEN (CRT,E)
4800 GOTO 140
4810 REM
4820 #""
4830 REM P9 HOLDS PRINTER LINE LENGTH.  INITIALLY SET TO CRT LINE LENGTH
4840 P9=63
4850 INPUT "Is report format on file ",C$
4860 REM IF REPORT FORMAT ON FILE GOTO GET REPORT FORMAT ROUTINE
4870 IF C$="Y" OR C$="y" THEN Z1=1 : GOSUB 6480
4880 IF Z1=1 THEN 5180
4890 IF S2=1 THEN INPUT "Enter printer line length ",P9
4900 J1=1
4910 FOR I1=1 TO Z
4920   K1=J1+L(I1)-1
4930   R9=0
4940   #"Label";%2I;I1;"  ";E$(J1,K1)
4950   INPUT "Enter label report line and position ",R9,P8
4960   R9=R9+.01*P8
4970   IF Z0=1 THEN R8(K3)=R9 : Z0=0 : GOTO 4990
4980   R8(I1)=R9 : L8(I1)=I1
4990   GOSUB 6040 : REM CALL SORT ROUTINE
5000   D1=0
5010   FOR K2=1 TO I1
5020     IF INT(R8(K2))<>INT(R8(K2-1)) THEN D1=0
5030     D1=D1+D(K2)+L(K2)+3
5040     IF D1>P9 THEN #"Line too long " : EXIT 5070
5050     NEXT K2
5060   GOTO 5110
5070   FOR K3=1 TO I1
5080     IF R8(K3)=R9 THEN EXIT 5100
5090     Z0=1 : NEXT K3
5100   GOTO 4930
5110   J1=K1+1
5120   NEXT I1
5130 INPUT "Do you want to save report format ",C$
5140 IF C$="Y" OR C$="y" THEN GOSUB 6250
```

```
5150  #""
5160  IF S2=1 THEN CURSOR 7,0 : #"Output is being printed"
5170  IF S2=1 THEN CLOSE (CRT,E) : OPEN (PRINTER,E)
5180  REM ************  REPORT GENERATION AREA  ***************
5190  REM
5200  REM Z3 HOLDS NUMBER OF RECORDS IN DATA BASE
5210  REM W8 HOLDS THE NUMBER OF RECORDS OUTPUT
5220  REM B1 HOLDS THE NUMBER OF RECORDS PER BLOCK
5230  REM W9 HOLDS NUMBER OF LINE TO BE PRINTED BETWEEN RECORDS
5240  REM D$,G$ THESE STRINGS GET VALUES FROM SELECTION ROUTINE
5250  B2=INT(B0/B1) : A=B0-B1*B2 : B2=B2+4 : IF A>0 THEN B2=B2+1
5260  IF A=0 THEN A=B1
5270  E2=INT(E0/B1) : E3=E0-B1*E2 : E2=E2+4 : IF E3>0 THEN E2=E2+1
5280  IF S2=1 THEN OPEN (PRINTER,E)
5290  W8=0
5300  FOR Z1=B2 TO E2
5310    GET (1,E,A$,Z1)
5320    IF E>=1 THEN EXIT 7160
5330    FOR J9=A TO B1
5340      W8=W8+1 : IF W8=E0+1 THEN EXIT 5660
5350      S0=0 : IF D$="ALL" THEN 5460
5360      FOR I2=1 TO Z
5370        J1=0 : D1=0 : K1=0
5380        FOR I9=1 TO L8(I2)
5390          J1=K1+1 : K1=J1+L(I9)-1 : D1=D1+D(I9)
5400          NEXT I9
5410        U=D(L8(I2))
5420        IF G$(D2-(D(I3)-1),D2)=A$(J9*D-D+(D1-(U-1)),J9*D-D+D1) THEN S0=1
5430        IF S0=1 THEN EXIT 5460
5440        NEXT I2
5450      GOTO 5650
5460      FOR I1=1 TO Z
5470        J1=0 : D1=0 : K1=0
5480        FOR I9=1 TO L8(I1)
5490          J1=K1+1
5500          K1=J1+L(I9)-1
5510          D1=D1+D(I9)
5520          NEXT I9
5530        IF INT(R8(I1))<>INT(R8(I1-1)) THEN #" "
5540        REM ANY LINE NUMBER GREATER THAN 14 IS IGNORED FOR OUTPUT
5550        IF INT(R8(I1))>14 THEN 5580
5560        U=D(L8(I1))
5570        #E$(J1,K1);" ";A$(J9*D-D+(D1-(U-1)),J9*D-D+D1);"  ";
5580        NEXT I1
5590      IF S2=0 THEN 5630
5600      FOR W0=1 TO W9
5610        #" "
5620        NEXT W0
5630      IF S2=0 THEN #" "
5640      IF S2=0 THEN INPUT "Press RETURN to continue ",C$
5650      NEXT J9
5660    NEXT Z1
5670  CLOSE (PRINTER,E)
5680  OPEN (CRT,E)
5690  S2=0
5700  GOTO 140
5710  REM ******************  DELETE RECORDS ROUTINE  **************
5720  #""
5730  INPUT "Enter record number to be deleted ",R0
5740  IF R0>Z3 THEN #"Exceeds number of records in file " : GOTO 5730
5750  IF R0<0 THEN #"There are no negative records " : GOTO 5730
5760  B=INT(R0/B1)
5770  A=R0-B*B1
5780  IF A>0 THEN B=B+1
5790  B=B+4
5800  GET (1,E,A$,B)
5810  IF E>=1 THEN 7160
5820  IF A=0 THEN A=B1
5830  J=A
5840  REM   S1=1  TELLS SCREEN PRINT ROUTINE THAT DATA IS TO BE OUTPUT
5850  REM   S7=1  TELLS SCREEN PRINT ROUTINE TO PRINT ONLY ONE RECORD
5860  S1=1 : S7=1
5870  GOSUB 2310 : REM CALL SCREEN PRINT ROUTINE
5880  CURSOR 15,0
5890  #"                                                          ";
5900  CURSOR 15,0
5910  INPUT1 "Delete record (Y/N) ",C$
5920  IF C$="Y" OR C$="y" THEN 5940
5930  GOTO 5970
5940  A$(J*D-(D-1),J*D)=F$(J*D-(D-1),J*D)
5950  PUT (1,E,A$,B)
5960  IF E>=1 THEN 7160
5970  CURSOR 15,0
5980  #"                                                          ";
5990  CURSOR 15,0
6000  INPUT1 "Another to be deleted ",C$
6010  IF C$="Y" OR C$="y" THEN 5710
6020  S1=0 : S7=0
6030  GOTO 140
6040  REM **************  SORT ROUTINE  **********************
6050  REM  SORT ROUTINE WAS TAKE FROM "A COMPARISON OF SORT ROUTINES"
6060  REM BY JOHN P. GRILLO, CREATIVE COMPUTING NOV-DEC 1976
6070  Y=I1
6080  M6=I1
6090  M6=INT(M6/2)
6100  IF M6=0 THEN 6240
6110  K6=Y-M6
6120  J6=1
6130  I6=J6
6140  L6=I6+M6
6150  IF R8(I6)<=R8(L6) THEN 6210
6160  T(1)=R8(I6) : T(2)=L8(I6)
6170  R8(I6)=R8(L6) : L8(I6)=L8(L6)
6180  R8(L6)=T(1) : L8(L6)=T(2)
6190  I6=I6-M6
6200  IF I6>1 THEN 6140
6210  J6=J6+1
6220  IF J6>K6 THEN 6090
6230  GOTO 6130
6240  RETURN
6250  REM *****************  SAVE REPORT FORMAT ROUTINE  *************
6260  INPUT "Enter output device number ",F
6270  IF F<0 OR F>1 THEN 6260
6280  INPUT "Enter report file name ",B$
6290  B$=B$+".FT"+CHR$(0)
6300  OPEN (3,E,B$,1,F)
6310  A$=F$
6320  FOR I1=1 TO Z
6330    T0=R8(I1)*100
6340    CONVERT T0 TO C$(####)
6350    A$(I1*4-3,I1*4)=C$
6360    NEXT I1
6370  CONVERT P9 TO C$(###)
6380  A$(241,243)=C$
6390  PUT (3,E,A$,0)
6400  A$=F$
6410  FOR I1=1 TO Z
6420    CONVERT L8(I1) TO C$(##)
6430    A$(I1*2-1,I1*2)=C$
6440    NEXT I1
6450  PUT (3,E,A$,1)
6460  CLOSE (3,E)
6470  RETURN
6480  REM **************  GET REPORT FORMAT ROUTINE  **************
6490  INPUT "Enter input device number ",F
6500  INPUT "Enter report file name ",B$
6510  B$=B$+".FT"+CHR$(0)
6520  OPEN (3,E,B$,2,F)
6530  GET (3,E,A$,0)
6540  FOR I1=1 TO Z
6550    C$=A$(I1*4-3,I1*4)
6560    CONVERT C$ TO R8(I1)
6570    R8(I1)=R8(I1)*.01
6580    NEXT I1
6590  C$=A$(241,243)
6600  CONVERT C$ TO P9
6610  GET (3,E,A$,1)
6620  FOR I1=1 TO Z
6630    C$=A$(I1*2-1,I1*2)
6640    CONVERT C$ TO L8(I1)
6650    NEXT I1
6660  CLOSE (3,E)
6670  RETURN
6680  REM  *************  PURGE BLANK RECORDS ROUTINE  ***************
6690  REM
6700  REM THIS ROUTINE SEARCHES THE FILE FOR BLANK RECORDS AND STORES
6710  REM THEIR LOCATION IN ARRAY B.  NONBLANK RECORDS ARE THEN
6720  REM TAKEN FROM THE END OF THE FILE TO FILL IN THE BLANKS
6730  REM
6740  A$=" " : FOR I=1 TO 8 : A$=A$+A$ : NEXT I : F$=A$ : T$=F$
6750  B=INT(Z3/B1)
6760  A=Z3-B*B1
6770  IF A>0 THEN B=B+1
6780  B=B+5 : K1=0 : Z0=0
6790  FOR I1=5 TO B
6800    GET (1,E,A$,I1)
6810    FOR J1=1 TO B1
6820      IF Z0=Z3 THEN EXIT 6860
6830      Z0=Z0+1
6840      IF A$(J1*D-(D-1),J1*D)=F$(D-(D-1),D) THEN K1=K1+1 : B(K1)=Z0
6850      NEXT J1
6860    NEXT I1
6870  FOR I1=1 TO K1
6880    B2=INT(B(I1)/B1)
6890    A2=B(I1)-B2*B1
6900    IF A2>0 THEN B2=B2+1
6910    B2=B2+4
6920    IF A2=0 THEN A2=B1
6930    B=INT(Z3/B1)
6940    A=Z3-B*B1
6950    IF Z3<B(I1) THEN EXIT 7110
6960    IF A>0 THEN B=B+1
6970    B=B+4
6980    IF A=0 THEN A=B1
6990    GET (1,E,A$,B)
7000    IF A$(A*D-(D-1),A*D)=F$(D-(D-1),D) THEN Z3=Z3-1 : GOTO 6930
7010    IF A$(A*D-(D-1),A*D)=F$(D-(D-1),D) AND B(I1)=Z3 THEN Z3=Z3-1 : EXIT 7110
7020    Z3=Z3-1
7030    T$(A*D-(D-1),A*D)=A$(A*D-(D-1),A*D)
7040    A$(A*D-(D-1),A*D)=F$(D-(D-1),D)
7050    PUT (1,E,A$,B)
7060    GET (1,E,A$,B2)
7070    A$(A2*D-(D-1),A2*D)=T$(A*D-(D-1),A*D)
7080    PUT (1,E,A$,B2)
7090    NEXT I1
7100  REM UPDATE RECORD ZERO FOR NEW RECORD COUNT
7110  GET (1,E,A$,0)
7120  CONVERT Z3 TO C$(###)
7130  A$(250,252)=C$
7140  PUT (1,E,A$,0)
7150  GOTO 140
7160  REM *****************  DISK ERROR ROUTINES  ****************
7170  IF E=1 THEN #"FILE NOT FOUND. "
7180  IF E=2 THEN #"NOT ENOUGH ROOM ON DISKETTE"
7190  IF E=3 THEN #"DUPLICATE FILE NAME ON DISKETTE"
7200  IF E=4 THEN #"FILE NOT OPEN "
7210  IF E=5 THEN #"END OF FILE ENCOUNTERED"
7220  IF E=6 THEN #"FILE ALREADY OPENED"
7230  IF E=9 THEN #"SYSTEM ERROR "
7240  IF E=12 THEN #"TOO MANY OUTPUT FILES ON ONE DRIVE"
7250  IF E>0 THEN #"ALL FILE HAVE BEEN CLOSED"
7260  IF E>0 THEN CLOSE (0,E) : CLOSE (1,E) : CLOSE (2,E) : CLOSE (3,E)
7270  IF E>0 THEN #"ALL PROCESSING STOPPED"
7280  END
7290  REM **************  END OF PROCESSING  *****************
7300  CLOSE (0,E)
7310  CLOSE (1,E)
7320  CLOSE (3,E)
7330  #"End of processing"
7340  END
7350  REM ******** INCORRECT NUMERIC FIELD INFORMATION ROUTINE********
7360  CURSOR 15,0
7370  #"                                                          ";
7380  CURSOR 15,0
7390  IF S0=1 THEN #"Field length exceeded "; : S0=0 : GOTO 7410
7400  #"Numeric information only please";
7410  CURSOR R(L),C(L)+L(L) : #" ";
7420  IF U8=1 THEN 7480
7430  FOR M1=1 TO D(L)
7440    #"x";
7450    NEXT M1
7460  #"  ";
7470  CURSOR R(L),C(L)+L(L)
7480  FOR U7=1 TO 225 : # : NEXT U7
7490  U8=0
7500  RETURN
```

Multiple statements on a single line following an IF statement execute when the IF statement tests true and doesn't terminate with an ELSE condition. If the statement is false and not terminated with an ELSE, they will be ignored.

Relieve SWTP Tape Drudgery

Generate MIKBUG-compatible tapes with the D2 kit.

Dr. Bernard Tan
Dept. of Physics
University of Singapore
Singapore 10

When hand-loading a machine-language program using a keyboard into a 6800 system such as the SWTP, you are tying up your system. Many owners of the SWTP and other 6800 systems also possess an MEK 6800D2 kit, which has a hex keypad. The D2 kit has a Kansas City Standard cassette interface, which reads and generates tapes in a binary format not compatible with the usual MIKBUG "S" format.

The accompanying program, D2MIKPUN, enables the D2 kit to generate MIKBUG-format tapes. In this way, you can use the D2 kit to hand-load machine code, which can be dumped onto a cassette for subsequent loading into the main SWTP or other 6800 system.

The program also puts onto the MIKBUG tape an address, which can be specified by the user and which may be different from that of the D2 kit memory in which the code resides. In this way, you can hand-load blocks of code into the D2 and dump them on tape with successive blocks of addresses meant for the larger 6800 system.

How It Works

The program makes use of the D2's JBUG routine and occupies 128 bytes from hex 0000 to 007F. The D2 comes with 256 bytes of RAM; the next 128 bytes from hex 0080 to 00FF may be used for loading the machine code. A long program several kilobytes long may be loaded and dumped on tape in 128-byte blocks with the correct addresses for the larger 6800 system.

To use the program, load it into the D2 manually. The program may then be saved on tape in the D2 binary format for future use. The desired machine code is then loaded manually into the remaining D2 RAM.

To generate the MIKBUG tape, the following RAM locations must be loaded (all addresses in hex):

A000—high byte of beginning address on MIKBUG tape
A001—low byte of beginning address on MIKBUG tape
A002—high byte of beginning address in D2 memory
A003—low byte of beginning address in D2 memory
A004—high byte of ending address in D2 memory
A005—low byte of ending address in D2 memory

On executing a GO to the start of D2MIKPUN at hex 0003, the MIKBUG tape is punched. One nice feature is that after punching, the beginning address in hex A000/A001 for the next block, if contiguous, is set to the correct value. With this program the D2 becomes an off-line loading terminal for a larger 6800 system. ■

D2MIKPUN program.

```
                   NAM    D2MIKPUN
              *
              * OFFSET MIKBUG TAPE PUNCH PROGRAM
              * FOR MEK6800D2 KIT
              * BY B.T.G. TAN, UNIVERSITY OF SINGAPORE
              * THIS PROGRAM ALLOWS THE D2 KIT TO PUNCH
              * MIKBUG FORMAT TAPES WITH BEGINNING
              * ADDRESS ON TAPE DIFFERENT FROM THAT OF
              * THE D2 MEMORY WHOSE CONTENTS ARE BEING
              * PUNCHED.  ADDRESSES TO BE STORED AS FOLLOWS:
              * $A000 AND $A001: BEGINNING ADDRESS TO
              *                  BE PUNCHED
              * $A002 AND $A003: BEGINNING ADDRESS IN D2
              *                  MEMORY
              * $A004 AND $A005: ENDING ADDRESS IN D2 MEMORY
              *

                     OPT    NOP
8008          ACIAC  EQU    $8008      ACIA CONTROL REGISTER
E0AC          INIT   EQU    $E0AC      JBUG CONTROL
E37A          OUTCH  EQU    $E37A      JBUG OUTPUT SUBROUTINE
E387          PUN    EQU    $E387      JBUG PUNCH SUBROUTINE
A000                 ORG    $A000
A000          PBEGA  RMB    2          PUNCH BEGINNING ADDRESS
A002          BEGA   RMB    2          MEMORY BEGINNING ADDRESS
A004          ENDA   RMB    2          MEMORY END ADDRESS
0000                 ORG    0
0000          TW     RMB    2          TEMPORARY BEGINNING ADDRESS
0002          TEMP   RMB    1          BYTE COUNT

              * PROGRAM STARTS HERE
0003 86 51           LDA A  #$51       INITIALIZE ACIAC
0005 B7 80 08        STA A  ACIAC
0008 FE A0 02        LDX    BEGA
000B DF 00           STX    TW         STORE TEMP BEGINNING ADDRESS
000D B6 A0 05 PUNCH1 LDA A  ENDA+1
0010 90 01           SUB A  TW+1
0012 F6 A0 04        LDA B  ENDA
0015 D2 00           SBC B  TW
0017 26 04           BNE    PUNCH2
0019 81 10           CMP A  #$10
001B 25 02           BCS    PUNCH3
001D 86 0F    PUNCH2 LDA A  #$F
001F 4C       PUNCH3 INC A
0020 97 02           STA A  TEMP       STORE BYTE COUNT
              * PUNCH CR, LF, 4 NULLS, S1
0022 CE 00 78        LDX    #MTAPE
0025 8C 00 80 PDATA  CPX    #MTAPE+8
0028 27 05           BEQ    PUNCH4
002A BD E3 87        JSR    PUN        JUMP TO JBUG PUNCH SUBR
002D 20 F6           BRA    PDATA
              * PUNCH FRAME COUNT
002F 96 02    PUNCH4 LDA A  TEMP
0031 8B 03           ADD A  #$3
0033 16              TAB               INITIALIZE CHECKSUM
0034 8D 2C           BSR    PTAPE2     PUNCH FRAME COUNT
              * PUNCH BEGINNING ADDRESS
0036 CE A0 00        LDX    #PBEGA
0039 8D 23           BSR    PTAPE1     PUNCH HIGH BYTE
003B 8D 21           BSR    PTAPE1     PUNCH LOW BYTE
              * PUNCH DATA
003D DE 00           LDX    TW
003F 8D 1D    PUNCH5 BSR    PTAPE1     PUNCH ONE BYTE (2 FRAMES)
0041 7C A0 01        INC    PBEGA+1    INCREMENT PBEGA
0044 26 03           BNE    PUNCH6
0046 7C A0 00        INC    PBEGA
0049 7A 00 02 PUNCH6 DEC    TEMP       DECREMENT BYTE COUNT
004C 26 F1           BNE    PUNCH5
004E DF 00           STX    TW
0050 53              COM B
0051 17              TBA
0052 8D 0E           BSR    PTAPE2     PUNCH CHECKSUM
0054 09              DEX
0055 09              DEX
0056 BC A0 04        CPX    ENDA       END ADDRESS ALREADY?
0059 26 B2           BNE    PUNCH1     NO
005B 7E E0 AC        JMP    INIT       YES, JUMP TO JBUG CONTROL

              * PUNCH 2 HEX CHARACTERS
005E EB 00    PTAPE1 ADD B  0,X        UPDATE CHECKSUM
0060 A6 00           LDA A  0,X        LOAD DATA BYTE IN ACIA
0062 36       PTAPE2 PSH A
0063 8D 05           BSR    OUTHL      OUTPUT LEFT HEX CHAR
0065 32              PUL A
0066 8D 05           BSR    OUTHR      OUTPUT RIGHT HEX CHAR
0068 08              INX
0069 39              RTS
006A BD E2 9F OUTHL  JSR    $E29F      JUMP TO JBUG SHIFT SUBR
006D 84 0F    OUTHR  AND A  #$F        OUTPUT ONE HEX CHAR
006F 8B 90           ADD A  #$90
0071 19              DAA
0072 89 40           ADC A  #$40
0074 19              DAA
0075 7E E3 7A        JMP    OUTCH      JUMP TO JBUG OUTPUT SUBR
0078 0D       MTAPE  FCB    $0D,$0A,0,0,0,0,'S,'1
                     END

NO ERROR(S) DETECTED
```

How's your love life?

A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stimulating computer game ever conceived. It combines a computer interview, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!

Interlude is: romantic . . . playful . . . outrageous . . . a fantasy. Interlude is: ■ Wet fun on a hot summer night. (Interlude #21) ■ A surprise on the way home from dinner. (Interlude #42) ■ A bubble bath that ends with a bang. (Interlude #78) ■ An evening to rest while she does all the "work." (Interlude #25) ■ The most romantic of evenings. (Interlude #84) ■ A new twist to an old subject. (Interlude #69) ■ Just watching her . . . (Interlude #57) ■ An erotic fantasy! (Interlude #33)

With over 100 Interludes, you can satisfy all levels of interest and desire. Each Interlude is fully described in the manual, and the more elaborate ones are detailed with regard to settings, props, and mood-enhancing techniques. But we've saved a few super Interludes for that very special time when your interview indicates you're ready! At that time, you will be introduced to one of several Interludes held secret within the computer. (When you learn secret Interlude #99, your love life may never be the same again!) Interlude can give you experiences you'll never forget. Are you ready for it?

Interlude™
The Ultimate Experience.

Interlude, 10428 Westpark, Houston, Texas 77042 I'm really ready! Rush me ____ copies of Interlude today.
☐ For the Apple II (16K) # ☐ For the TRS-80 (Level II-16K) ## ☐ $14.95 for cassette ☐ $17.95 for diskette.
Add $1.50 for shipping. Texas residents add 6% sales tax. My check (payable to Interlude) is enclosed. ☐
*Charge my ☐ MASTERCHARGE ☐ VISA account.
Account No. ____ Expiration date ____
All charge customers must sign. Signature ____
Name ____ Age ____
Address ____
City ____ State ____ Zip ____

*CHARGE CUSTOMERS: Order by phone toll-free! **1-800-327-9009 Ext. 306**

Apple II is a registered trademark of Apple Computers, Inc. ## TRS-80 is a registered trademark of Radio Shack, a Tandy Co.

Dynamic Duo

Any problem has at least two good solutions—to wit, the following pair of articles addressing the SWTP-Heath H14 interface question. The first uses a modified MP-S board . . .

Dr. P. Vijlbrief
Dept. of Radiology
University of Leiden
The Netherlands

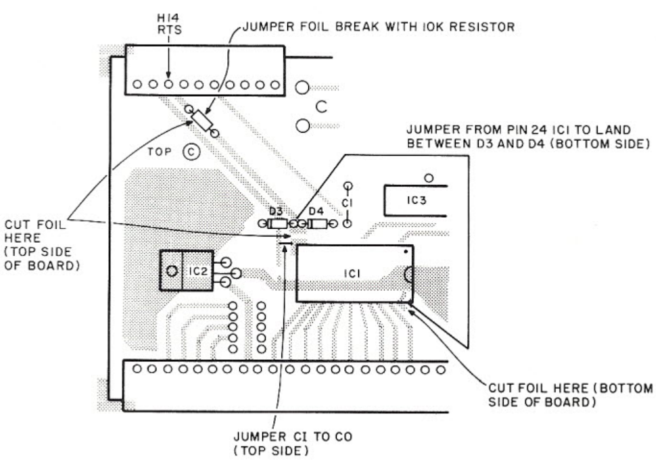


Fig. 1. MP-S board with modifications.

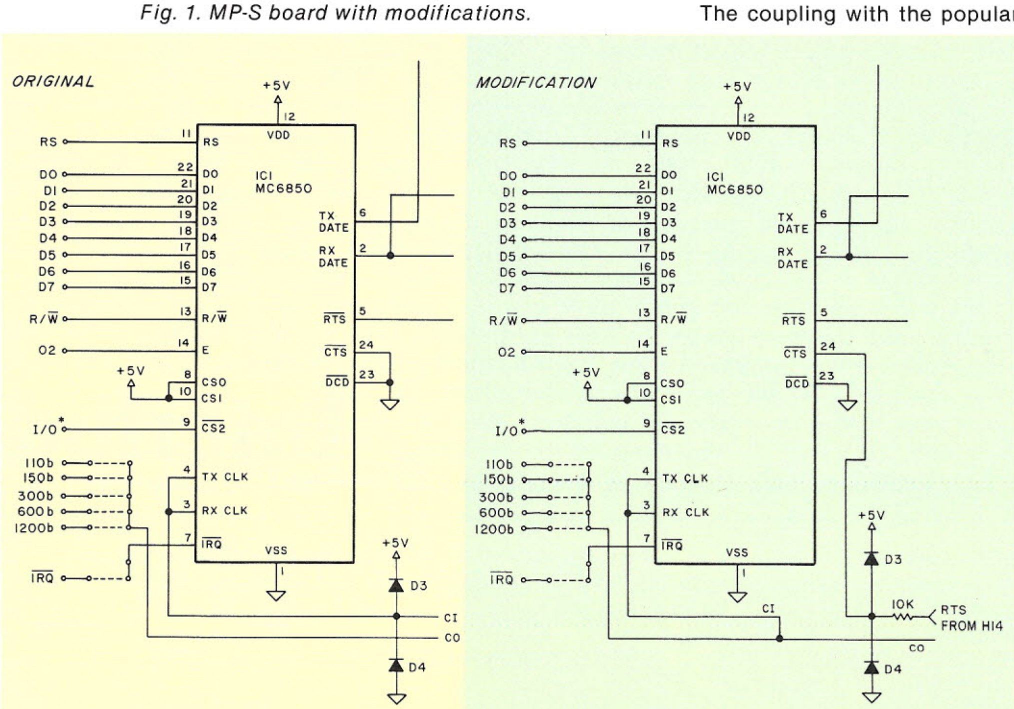


Fig. 2. Detail of MP-S serial interface schematic.

The popularly priced H14 line printer is a good buy. The kit comes complete—it can be assembled in about 25 hours. The coupling with the popular SWTP system was quick and easy.

Modifications

I used an MP-S serial interface board. Only a few small modifications were necessary to handle the handshaking feature of the H14 printer. The RTS (ready to send) signal coming out of the printer should be connected to the I/O card. Therefore, I used the CI connection. I cut the foil between the point where D3 and D4 meet and the connection to the ACIA. The ACIA side of the foil break is jumpered with a small piece of wire to a nearby point on the CO foil to make sure that the clock signal is routed to the ACIA.

I broke the grounding foil coming from pin 24 of the ACIA ($\overline{CTS}$) by carefully using a sharp knife, but leaving pin 23 connected to ground. Then I soldered an isolated wire between pin 24 of the ACIA and the joint between D3 and D4. Furthermore, I broke the foil coming from the top board connector point CI to D3 and D4 and jumpered the break with a 10k Ohm resistor. The modifications are clearly seen in Fig 1. Fig. 2 shows a part of the schematic of the MP-S board, both the original and the modified circuit.

From the H14 interface cable the protective ground (PGND) pin 1 and the signal ground

Hex code	Mnemonic	Description
37	PSH B	save contents of ACC B
C6 03	LDA B #$ 03	
F7 800C	STA B $ 800C	ACIA address
C6 11	LDA B #$ 11	
F7 800C	STA B $ 800C	
33	PUL B	restore contents of ACC B
39	RTS	back to main program

Example 1. Initialization subroutine for ACIA in port 3.

Hex code	Mnemonic	Description
37	PSH B	save contents of ACC B
FF A060	STX $ A060	save contents of X-register
CE 800C	LDX #$ 800C	load X-register with ACIA address
E6 00	LDA B 0,X	check ACIA ready?
57	ASR B	
57	ASR B	
24 FA	BCC CHECK AGAIN	
A7 01	STA A 1,X	write contents of ACC A to ACIA
FE A060	LDX $ A060	restore contents of X-register
33	PUL B	restore contents of ACC B
39	RTS	back to main program

Example 2. Subroutine to output contents to ACC A to printer via ACIA in port 3.

(SGND) pin 7 should be connected to the ground point on the MP-S I/O connector. The RS-232C serial input (SIN) pin 3 goes to the RO point on the MP-S I/O connector, and the request to send (RTS) output pin 4 goes to MP-S I/O connector point CI (now $\overline{\text{CTS}}$). Only RI must be jumpered to ground on the MP-S I/O connector. Do not jumper TI to TC (see Fig. 3).

Programs

If you put the MP-S board in I/O port 3 of your SWTP system, you can use the H14 printer without further problems with SWTP 8K BASIC. Jumper the H14 for RS-232C input and output. If you want to use the printer with any other program, you first have to initialize the ACIA in the beginning of the program and then use a typical ACIA output program to output the contents of ACC A to the printer.

Examples 1 and 2 show the programs I used for the MP-S board in port 3, but you can easily change to other ports by replacing the ACIA address in the program (port 3 address is $800C).

If you want to change the character width of the printer through software, be careful to load the needed control characters one after another in hex values in ACC A and output them via the output subroutine mentioned earlier. Simply loading the characters in ACIA address + 1 is not the right way to do this because the ACIA has to be checked first.

I hope that you will be as satisfied as I am with the SWTP-Heathkit combination. ■

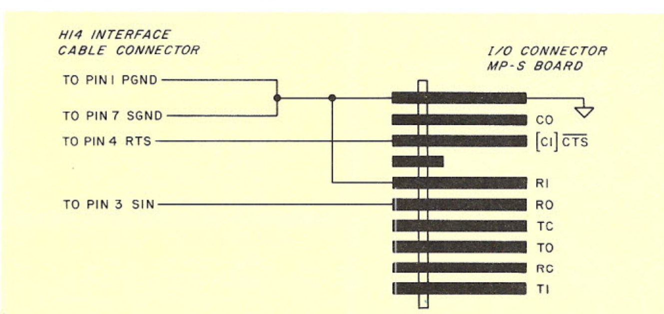


Fig. 3. Interface connection.

✓ Reader Service—see page 258

David J. Graves
Associate Professor
Dept. of Chemical & Biochemical Engineering
University of Pennsylvania
311A Towne Bldg. 220 S. 33rd St.
Philadelphia PA 19104

Dynamic Duo (cont.)

. . . while this one describes an entirely home-brew interface.

The Heath H14 Printer is an attractive way of providing hard-copy output at a low cost. Its speed, use of economical standard paper in a range of widths and variety of print densities are a few of the reasons why I chose it for my system. The major drawback is that it is designed to operate with a serial interface at a recommended baud rate of 4800 with special control characters, while most (non-Heath) microcomputers are designed to operate printers from a parallel port with handshaking control signals.

Interfacing the H14 with such a microcomputer can be done in two ways: a parallel input-serial output port with handshaking control can be constructed; or an ordinary serial port can be used, and the printer driver software routines can be rewritten to recognize the control signals generated by the H14 and respond in the appropriate fashion.

I have used the first approach in interfacing the H14 to a Southwest Technical Products 6800 microcomputer. It has the advantage that it could be applied to almost any microcomputer system, particularly one with a 6821 (or the equivalent 6820) peripheral interface adapter (PIA) as the parallel port. It also requires little or no software modification, which may be an important consideration if a source listing for the printer routines is not available.

The second approach, using software, may, in some cases, be simpler where an uncommitted serial port is available and the printer routines can be modified easily.

Design Details

The SWTP 6800 outputs data to a parallel PIA port by placing it on the bus lines and then sending an active low strobe pulse on CA2 (pin 39 of U1, Fig. 1). When the external device has accepted the data, it responds with a similar low pulse on CA1 (pin 40). These pulses are called "handshaking" signals and are labeled $\overline{\text{DATA}}$ $\overline{\text{AVAILABLE}}$ and $\overline{\text{DATA}}$ $\overline{\text{ACCEPTED}}$, respectively (see Fig. 2).

A similar protocol is followed in many computers, even if a device other than a PIA is used in the interface. In fact, if you are clever enough, you can substitute other interface hardware such as the 8255 programmable peripheral interface, or even omit the PIA entirely and feed the parallel data bits directly into a universal asynchronous receiver/transmitter (UART, U2).

In my case, omitting the PIA was undesirable because it would have destroyed the software compatibility and plug-in simplicity I wished to preserve. All of the data bus signals to the PIA are shown for the convenience of those using the SS-30 bus, but for other interfaces, you can simply supply the eight data bits and observe the handshaking protocol just outlined.

The data bits go from the PIA to the UART and are strobed in by CA2. A 4800 baud (16×) clock, which can be found on most computers, operates circuitry internal to the UART to send the data out as a serial stream through pin 25. The clock signal in the 6800 is on the CPU board, and Southwest Tech gives information on how to bring it out on the bus. Transistor Q1 and its associated circuitry convert the TTL voltage levels to RS-232C compatible values.

Before the $\overline{\text{DATA}}$ $\overline{\text{ACCEPTED}}$ pulse can be generated, several things must be true: 1) the UART

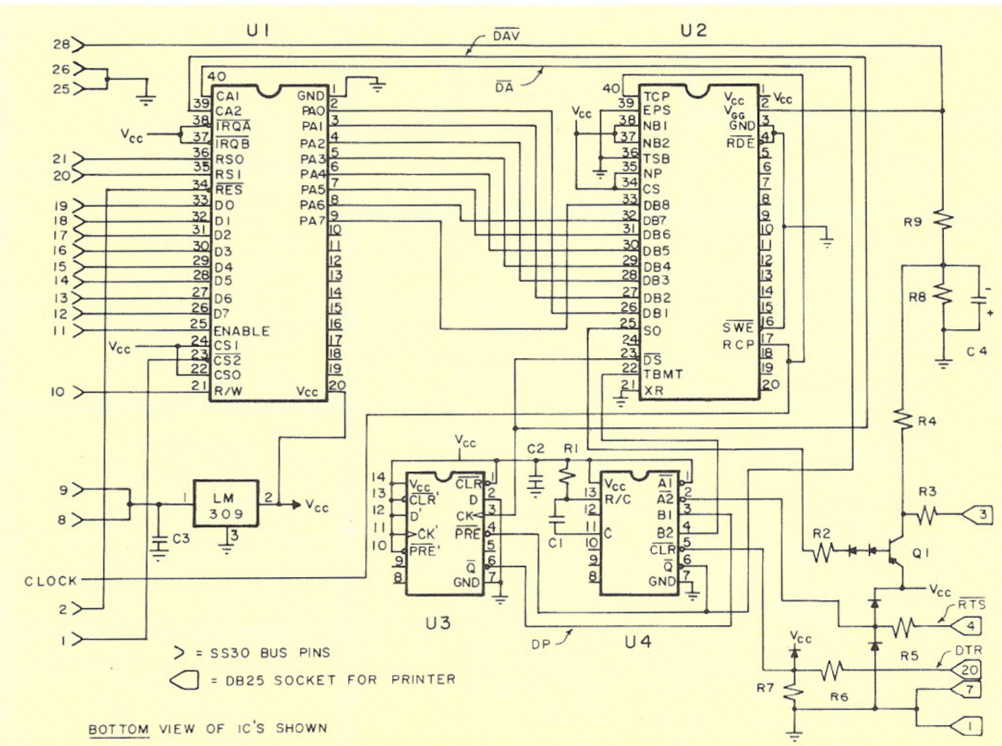


Fig. 1. Schematic of the printer interface circuit.

Resistors (all 1/4 W 5 percent)

R1	20k
R2, R6	1k
R3	200
R4	680
R5	2k
R7	375
R8, R9	430

Capacitors

C1	1000 pF	50 V ceramic
C2, C3	0.1 uF	50 V ceramic
C4	10 uF	16 V electrolytic

Diodes

All five diodes are type IN914 or IN4148

Transistor

Q1—2N2907

Integrated Circuits

U1	—Motorola MC6820 or MC6821 PIA
U2	—General Instrument AY-5-1013 UART or Western Digital TR-1602B or Texas Instruments TMS6011
U3	—7474 TTL dual D type flip-flop
U4	—74122 TTL retriggerable one-shot
LM309	—Voltage regulator, 5 V dc

Table 1. List of components.

Pin Number	Description
1,7	System power and signal grounds, respectively
3	Printer data in, RS-232C voltage levels
4	RTS, Request to Send Data (active low, RS-232C)
20	DTR, Data Terminal Ready (active high, RS-232C)

Table 2. H14 signals used in interface.

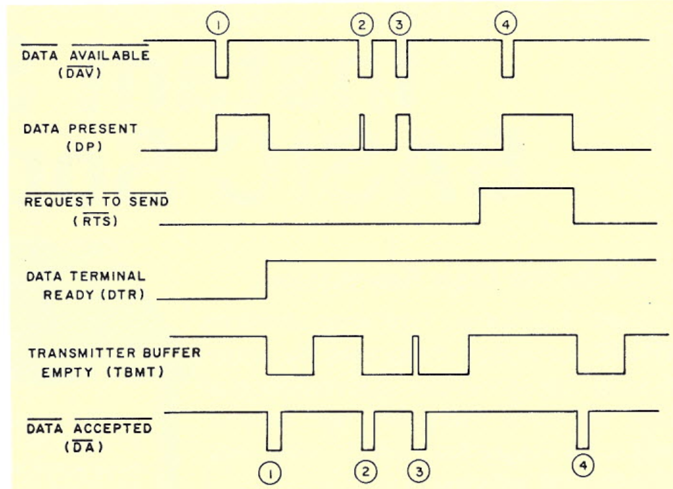


Fig. 2. The data accepted pulse is delayed from the data available pulse as shown for four conditions: 1) DTR initially false (printer off line); 2) no delay, all conditions true; 3) UART buffer full (TBMT false); 4) printer buffer full ($\overline{\text{RTS}}$ false).

transmitter buffer must be empty (signal TBMT); 2) the printer must be "on line" (signal DATA TERMINAL READY, DTR); 3) the data buffer of the printer must not be full (signal $\overline{\text{REQUEST TO SEND}}$, $\overline{\text{RTS}}$); and 4) the $\overline{\text{DATA AVAILABLE}}$ pulse must have occurred, clocking the flip-flop in U3 to latch on a high-level DATA PRESENT, DP, signal. The last of these conditions to become true (negative true for $\overline{\text{RTS}}$) triggers one-shot U4, which sends out the $\overline{\text{DATA ACCEPTED}}$ pulse to CA1. Fig. 2 shows several example cases of each of these signals controlling signal flow.

The only factor that may be critical if you omit the PIA and may differ from one computer to another is the length of this pulse. If it is too narrow, the computer will not see it, and if it is too wide, it may still be present when the computer returns with the next character, causing missing or garbled transmission.

Resistor R1 or capacitor C1 may be changed as necessary to shorten or lengthen the pulse. The values given produce a pulse about seven microseconds wide, which should be more than adequate for all but the slowest of systems. (The values shown should give you some output. If it is garbled, try reducing R1 or C1.)

A careful reader may think that this simple interface has a major flaw, since the first character in a data stream is always transmitted, even if one of the previous four conditions, particularly the printer being "on line," is not met. However, the H14 buffer still accepts characters in the off-line position, so when the on-line button is pushed, printing commences with no lost characters.

There are two conditions that will cause the first character to be lost when the computer tries to print: when the ac power to the printer is off or when the cables connecting printer and computer are unplugged. The additional circuitry needed to correct these rare operator errors did not appear to be justified.

One other point is worth noting: any of these three conditions (printer unplugged, off line or power off) will cause the computer to hang up in a loop waiting for the $\overline{\text{DATA ACCEPTED}}$ pulse, which never comes. In the SWTP 6800, striking Control C on the terminal will have no effect since the terminal port is not interrogated by the CPU until after it receives each expected pulse on the printer port. You should not hit RESET, as this will cause problems reentering the program. Simply put the printer on line and then, while it is printing, if you wish to regain command, type Control C. With these simple caveats, the printer should work perfectly.

The "on line" switch is convenient for temporarily interrupting output (for instance to reposition or replenish paper) with this interface. When released to the "off line" position, it will suspend operations at the end of the line currently being printed. When it is pressed again, printing resumes.

Construction Notes

The prototype model of this interface was constructed on perfboard with point-to-point wiring techniques. Parts placement is not critical, but C2 should be placed close to the supply pins of U3 and U4, and C3 should be close to the voltage regulator input pin. If there is enough interest from readers wishing to copy this interface, I will design and make available a printed circuit board to simplify construction. ■

Pin Number	Description
1	A decoded "$\overline{\text{port select}}$" signal (active low)
2	System $\overline{\text{reset}}$ (active low)
8, 9	System +8 V dc unregulated power
10	Read/$\overline{\text{write}}$ signal (high = read)
11	Phase 2 of system clock
12	Data bit 7
13	Data bit 6
14	Data bit 5
15	Data bit 4
16	Data bit 3
17	Data bit 2
18	Data bit 1
19	Data bit 0
20, 21	Address bits A1 and A0, respectively; selects one of four internal PIA registers
25, 26	System power and signal ground
28	system −12 V dc (regulated)
Clock	A clock signal 16X the 4800 baud transmission rate is not available on the bus but must be supplied (see text)

Table 3. SS-30 bus signals used in interface.

Instant Software: BASIC Sophistication

Join the Disk Revolution!

Instant Software has revolutionized TRS-80 business programs by introducing the speed and convenience of disk software. Here are two no-nonsense packages that will do your work rapidly and efficiently.

Our Energy Audit package is designed to save you money on home heating expenses. It will analyze your home and make specific recommendations for your insulation needs.

Our Accounts Receivable/Accounts Payable package is a versatile, easy-to-use business system. KB/Microcomputing magazine uses this system for all of its dealer book sales. It will soon be augmented by more business programs from the same author.

So stop going around in circles, take our disk software out for a spin. You won't be sorry.

ENERGY AUDIT

When it comes to spending money for energy to heat a home, everybody's a "skinflint." With today's rising fuel prices and staggering inflation, you need ways not only to cut down on your home heating bills but also to conserve our diminishing energy supplies.

One of the best ways to invest your money is to insulate your home. You'll save money, and your home will appreciate in value. But, you're asking yourself, "How do I start, where should I insulate, and what will it cost me?" The Energy Audit package can answer all these questions and more.

The programs in this package will give you detailed instructions on how to examine your home, collect information on its structural components and layout, and create a data file. The program will then analyze this data, show where you're losing money because of excessive heat loss, and describe the most cost-effective methods for saving energy and money.

You may select any area of your home for detailed scrutiny, input your own cost figures, calculate savings potentials for different options, and incorporate whichever you want into your own home-investment budget. In effect, you can create a computerized "model" of your home. You'll be able to see the changes that will be made in your energy budget as a result of adding a wood stove or refinishing an attic.

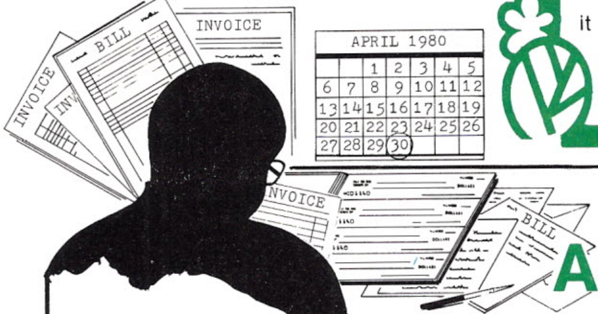
When your friends and neighbors see how much money you're saving, they will want their homes analyzed too. You could even start your own business, showing your customers exactly where and how to insulate their homes for maximum savings. The Energy Audit package is the perfect selling tool for any contractor or building supplies dealer.

With the Energy Audit package you'll know what your home needs, where it needs it, and how much it will cost you.

Or maybe you enjoy paying utility bills?

Pkg. 0089R (cassette version) $49.95.
Pkg. 0052RD (disk-based version) $75.00.

AVAILABLE ON FLOPPY DISKS



ACCOUNTS RECEIVABLE/ ACCOUNTS PAYABLE

Now, in one package, you can have a *complete* Accounts Receivable/Accounts Payable (AR/AP) system! These programs will handle all the drudgery involved in processing AR/AP entries.

Each program is capable of handling up to 760 accounts and as many as 1500 entries per month. Should you need to handle more accounts, you can divide them into as many sub-groups as necessary and keep a set of data disks for each sub-group.

The Accounts Receivable program can print invoices, statements, and address labels for each customer. It will also generate a Month End report, a Customer Activity report, and a Daily Sales report.

The Accounts Payable program will generate a Month End report, an Account report, a Daily Activity report and a Check Payment report.

The AR/AP package is ideal for any small business. The programs are self-prompting and are easily used by anyone familiar with AR/AP operations.

These programs can save you money, because they can print your company's letterhead at the top of each invoice and statement, using plain, fan-folded paper.

Accounts Receivable/Accounts Payable: Software for the Professional.

THIS PACKAGE REQUIRES THE FOLLOWING MINIMUM SYSTEM

1. A TRS-80 Level II microcomputer, 16K of memory.
2. An Expansion Interface with at least 16K of additional memory.
3. Three mini-disk drives.
4. A pin-feed line printer.
5. Any TRS-80 Disk Operating System.

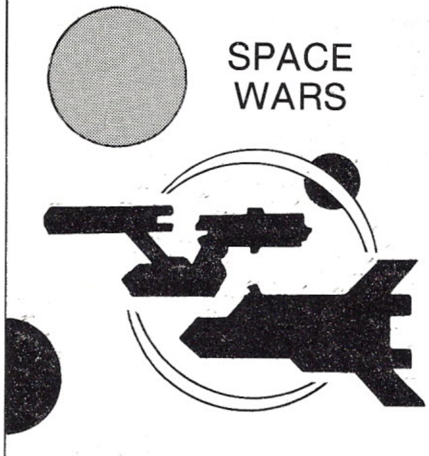
Ask for Package 0075RD **$199.95.**

Instant Software™ Inc. Peterborough, New Hampshire 03458 603-924-7296

Instant Software™ New Releases

40

APPLE*



SPACE WARS You must protect your planet against a wide variety of alien attacks. Included are:

- **Space War**—One or two players can pilot their saucers and duel with laser beams or shoot stars.
- **Star Ship Attack**—Your mission is to protect the food station satellites from destruction by the enemy star ship. You must capture, destroy, or drive off the attacking ship.
- **Battlefield**—Guess the location of the four enemy divisions and destroy them before your forces are wiped out.

Engaging in battle requires the Apple 8K and Integer BASIC. **Order No. 0096A $7.95.**

ACCOUNTING ASSISTANT

This package will help any businessman solve many of those day-to-day financial problems. Included are:

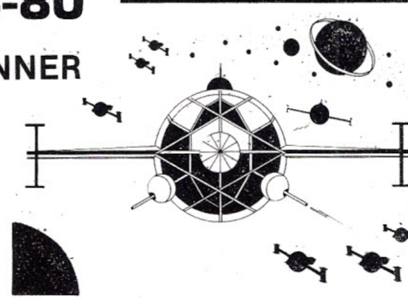
- **Loan Amortization Schedule**—This program will give you a complete breakdown of any loan or investment.
- **Depreciation Schedule**—You can get a depreciation schedule using any one of the following methods: straight line, sum of years-digits, declining balance, units of production, or machine hours.

This package requires the Apple 16K and Applesoft II BASIC. **Order No. 0088A $7.95.**

Featuring SOUND EFFECTS

TRS-80***

BALL TURRET GUNNER



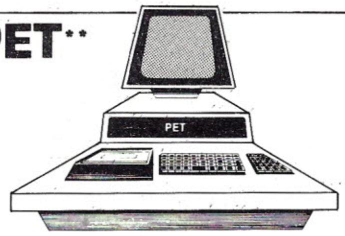
For years the Petro Resource Conglomerate has attacked our photon collection stations and strangled our deep-space trade routes. The PRC's Exxonerators Class light fighters (code name: Gnat) have been their main weapon. Now you can strike back, by joining the Ball Turret Gunner Service.

Imagine yourself at the control console of the LW-1417 Stratoblazer (Type B Strategic Laser Weapon). Your Hindsight Director informs you that a Gnat fighter is coming in for an attack. You position your gigawatt laser turret until you can see the target on your monitor. The Range Indicator shows he's coming in fast. You center your target sights on his location. The Targeting Computer studies his course and speed as your finger tenses over the firing key. You know you'll have only a fraction of a second in which to react. The Gnat fighter's evasive maneuvers cause him to dance in your sights. Suddenly you see the fire command, and you instinctively react. Your Stratoblazer's laser beam lashes out and reduces the Gnat fighter to an expanding ball of ionized gas. Mission accomplished.

Ball Turret Gunner, with your choice of multiple levels of difficulty, optional sound effects programming, and superb graphics, is more than just a game. It's an adventure. Experience it!

Order No. 0051R $9.95

PET**



PET UTILITY I

You're working under a serious handicap if you can't write programs in machine language. The PET Utility I package gives you the tools you need:

- **Monitr**—The Monitr program lets you write, edit, save, and verify any machine language and/or BASIC program. Just load and run the Monitr program and then load the program you want to edit.
- **Programmer's Calculator**—This program will convert numbers into the binary, octal, decimal, and hexadecimal systems and function as a floating point calculator. It will also display all four numbering systems simultaneously and allow you to handle large numbers limited only by the size of your screen.

For the 8K PET. **Order No. 0105P. $9.95.**

For a free catalog listing over 200 programs write: Instant Software Catalog Dept., Peterborough, N.H. 03458.

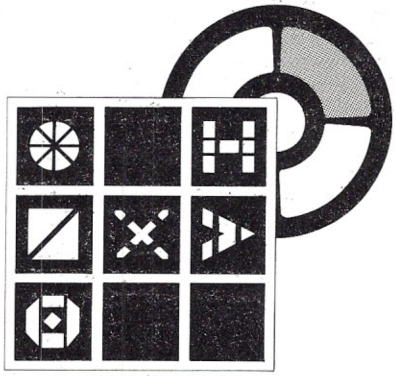
Look for Instant Software at a store near you. If the store nearest you does not stock Instant Software, use this order blank to purchase your software directly or call Toll-Free 1-800-258-5473.

*A trademark of Apple Computer Inc.
**A trademark of Commodore Business Machine Inc.
***A trademark of Tandy Corporation

Now available for TRS-80, PET and Apple.

MIMIC

A fast-action memory game.



In Mimic, players are shown a sequence of graphic characters that appear in random locations within a 3×3 grid. You'll have to remember the order and location in which the characters were shown, and "mimic" that sequence on your keyboard.

With several levels of skill, it's not hard to see why people of all ages enjoy this exercise of concentration and dexterity.

TRS-80 Level II 16K, **Order No. 0066R $7.95**
PET 8K, **Order No. 0039P $7.95**
Apple 24K (Integer), **Order No. 0025A $7.95**

Name ______________________

Address ______________________

City ______________________

State ____________ Zip ____________

☐ Check ☐ Money order

☐ VISA ☐ AMEX ☐ Master Charge

Card No. ______________________

Expiration Date ______________________

Signed ____________ Date ____________

Order your Instant Software today!

Quantity	Order No.	Unit Cost	Total Cost
		Handling	$1.00
		Total Order	

Instant Software Inc. DEPT. 40E0
Peterborough, N.H. 03458 USA

Instant Software™ Inc.

Peterborough, New Hampshire 03458 603-924-7296

Reader Service—see page 258

Program Property Rights

Program copyright is still a swamp of confusion. Here's advice on proving authorship.

Joseph G. Wackerman, JD
3204 Wendover Road
Troy MI 48084

The protection of your extensive efforts in creating a new program of which you are quite proud has been the subject of attempts to patent programs or segments of programs. The results have produced confusion, to say the least. Clarification is going to take years; but in the meantime, take heart. There is another way to protect your creation. In the field of law, sometimes referred to as idea theft, there is extensive case law giving ownership or property rights to ideas.

A patent, granted by the federal government, gives you ownership or property rights to the device or process. The subject is a physical entity. You own it, you can charge other people for the use of it and no one else can copy it. If the world were perfect, there would be no need for patents. You would be paid for your creativity by those who wished to use what you had developed.

The only reason for a patent is to protect you against unauthorized use—that is, the right to bring a lawsuit to recover for the unauthorized copying of your mechanical device or process.

Patenting Software—An Overview

The thinking about patenting software has been directed to the program listing. The embodiment of the creative thought that went into building the program has been thought of as a physically unique thing. For years, there has been a large body of law in the United States recognizing that ideas can also be protected. The thought process is, itself, protected.

If an idea is novel, unique and concrete, the courts will protect the creator in a situation where he has disclosed the idea in confidence or where the idea has been stolen. You do have the right to sue someone who has misappropriated your creativity to his own benefit. As in patent protection, the protection is the right to bring a lawsuit.

Any program listing is concrete. By concrete, I mean something that can be looked at and compared with that which is alleged to have been stolen. Concrete means that there is no ambiguity or vagueness in the expression of the idea. As everyone is well aware, there is no possibility for vagueness in a program listing.

The novelty or uniqueness of your creation is a question of fact; the novelty and, therefore, the uniqueness need only be in the eye of the user. If the unauthorized use exists, there is almost an admission that the user found this combination of computer expressions to be novel or unique to him. That is all that is needed.

It is not that rare when two minds will perfect the same idea—even at the same time. This does not destroy the uniqueness of any idea. Remember, we are after use in an unauthorized manner only. The important thing to keep in mind is that you are going after someone to whom you disclosed your idea.

A disclosure in confidence really means that when you showed someone this idea, in the form of a listing, you intended to sell it and were not merely showing off or giving it to the world freely. Presumably, a program of sufficient import to be worth a lawsuit would be of such size and seriousness that disclosure of it would undoubtedly be with the intention of selling it.

To avoid ambiguity before disclosing details, have an agreement that the disclosure is "in confidence" and for the purpose of negotiation. If you are a prospective buyer, you want the disclosure to be "not in confidence." The choice is a business one.

Should a situation ever arise where someone was bold enough to actually steal your program and attempt to benefit commercially from it, there is little doubt that the misappropriation of your work effort would give you the right to sue and recover money. No one is entitled to benefit from someone else's work. In this situation, you do not need all the elements I mentioned above—just prove you wrote it.

Specifics

So far, the discussion has been in generalities. The particulars of proof of authorship are more interesting. In writing a large program, you will undoubtedly amass a great deal of work papers. If these are dated and saved, they can be used not only to establish that you created the program, but their

sheer volume will go to establishing your effort. Mail the final listing to yourself, but do not open the envelope until the correct moment of the trial. This will give your lawyer fits—which is fun to do.

Of course, the best proof of your authorship would be to have the program announce to the world that you wrote it. On a microcomputer using BASIC, this can be a lot of fun. Programming in that language involves number and letter inputs with number or letter outputs.

As you go through life, there are innumerable number and letter combinations unique to yourself. The first three letters of your last name and the last two numbers of your street address are your laundry mark. You have social-security numbers, telephone numbers and zip codes, all tied to your life. You also have family members, where, again, you have letters and numbers related uniquely to yourself.

In even the shortest BASIC program, if there are no remarks explaining each line, it is relatively easy to bury in the program a tiny sub-program that can set the value of letter addresses to some number. After the program has been run, you then inquire of those memory addresses what number is contained in them. For example, my initials are JGW. After a run, if I inquire of one of my programs the numbers in those three addresses, I can produce my birthdate:

```
P. J, G, W
12    8    1926
```

You start with something as simple as

```
xxx     U=6
```

buried in a working line. Then later,

```
xx      J=U*2
xxx     W=(J*100)+(U*100)
```

and further down:

```
xxx     U=100:Z=26
xxx     G=4
```

Finally, at the end of the program:

```
xxx     G=G*2: W=W+U+Z
```

With these let statements buried in "working" lines, they can never be found.

On larger mainframes writing in languages less directly related to English, it should be no problem for a programmer to come up with some unique small program that would proclaim the name of the author. For instance, have a line that is not used but calls a second program buried in the listing. The second program prints the name.

Your valuable ideas can belong to you. Keep proof of your authorship and have the fun of hiding your trademark in the bits and bytes! ■

✓ Reader Service—see page 258

PUSH! POP! RAM! WOW!

Use this "quickie" to test your 8080's RAM.

Jack Dennon
Box G
Warrenton OR 97146

This program to test 8080 RAM is short, and it works. It is short enough—28 bytes—to toggle in from a switch panel. When I say it works, I mean it has isolated bad 2102s that passed that standby diagnostic test known as "deposit and examine."

This test is based on the 8080's PUSH and POP instructions. Evidently, these instructions place greater demand on memory circuits than do the MOV instructions.

The test starts at a user-specified high memory location and works toward lower memory addresses using a PUSH, PUSH, POP sequence. So load the program above whatever RAM you wish to test. The starting address (actually the first word address plus one) to be tested appears in the LXI SP instruction at relative locations 05 and 06. The test pattern is in relative location 01. Usually, I start with a pattern of alternate zeros and ones—55 or AA hex—and then if that turns up nothing, I try some other bit patterns.

Since only every other memory byte experiences a write-read sequence in consecutive instruction cycles, it is also worthwhile to try changing the starting address by one byte in the LXI SP instruction.

After the program stops, examine the contents of relative locations 1D and 1E at symbolic location ADDR, where the program deposits the address of the first mismatch. The test pattern is left stored in all tested locations, so by examining the contents of memory in the vicinity of the stop location, you will be able to determine what you have crashed into. It will be either a failed memory location or, if all RAM tested is OK, it will be a natural memory boundary.

In a non-failing, unbounded 64K RAM, the program would self-destruct. Also, with a test pattern of all bits on in a ROM-less system such as my Altair, where nonexisting memory reads out as all bits on, a non-failing memory causes the program to wrap its address space and push, push, pop right through itself, so watch out.

Use all bits on only as the last test pattern. The chaos following program destruction is then a sign that your memory is working just fine! ■

```
                    ;       THIS IS AN ACCESS TIME TEST.
                    ;       77/07/02.        J. DENNON.
                    ;
3F00                        ORG     3F00H
3F00 3E55           START:  MVI     A,55H
3F02 67                     MOV     H,A
3F03 6F                     MOV     L,A
3F04 31FF3E                 LXI     SP,3EFFH
3F07 E5             LOOP:   PUSH    H
3F08 E5                     PUSH    H
3F09 E1                     POP     H
3F0A BC                     CMP     H
3F0B C2123F                 JNZ     ERROR
3F0E BD                     CMP     L
3F0F CA073F                 JZ      LOOP
3F12 210000         ERROR:  LXI     H,0
3F15 39                     DAD     SP
3F16 221D3F                 SHLD    ADDR
3F19 C3193F         STOP:   JMP     STOP
3F1C 00                     NOP
3F1D 0000           ADDR:   DB      0,0
3F1F                        END     START
```

Program listing.

✓Reader Service—see page 258

Bruce Hoyt
Sharon Associate Reformed
Presbyterian Church
Rte. 1
Brighton TN 38011

Keeping Real Time with OSI's Superboard II

A real-time clock, with alarm and timer, for the 6502.

Recently I purchased a Superboard II from OSI. It is great in many ways, especially in price. But several times I have wished for a clock that could be used to set off an alarm. I have also wanted an interval timer that would provide an interrupt to be used for switching between programs and also to poll my keyboard. Frankly, I didn't want to go to all the trouble and expense of interfacing one of the many clock chips available. The following idea struck me like a bolt out of the blue.

Hardware Modification

You know, of course, that the 60 Hz power line frequency drives clocks of all kinds. This 60 Hz voltage could be squared up and brought down to TTL levels (i.e., 5 volts) and used to interrupt the 6502's NMI line. On the Superboard II there is an even easier way.

A 60 Hz square wave is generated by a divider chain driven from the clock and is used for the vertical sync in the video output circuit. To use it, all you need are two pieces of fine wire and one SPST switch. Connect one side of the switch to the 4.7k resistor R10, which pulls up the NMI line to 5 volts and is right in front of the "8" on the keyboard. The exposed lead on this resistor is the NMI interrupt line. Connect the other side of the switch to the C15 output coming from pin 11 on chip U61, which is a 74LS163.

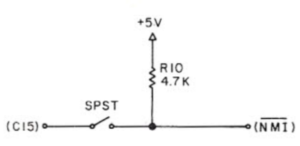


Fig. 1. Circuit diagram.

If you follow the printed circuit line from this pin an inch or so toward the front of the Superboard II, you will find a convenient place to which you may solder the other lead from the SPST switch. Be sure to use fine wire for this so as not to pull up the printed circuit lines. See Fig. 1 for a diagram of the circuit.

If you don't have a Superboard II, then you will need to use the 60 Hz power line frequency to drive the NMI interrupt line. Check Don Lancaster's TTL or CMOS cookbooks to learn how to do that.

The Program

The hardware is easy, and the software is almost as easy. The Superboard II NMI interrupt jumps out to \$0130. The memory from \$0130 to \$01C0 is unused, so I put my little clock routine there (see program listing). Since an interrupt occurs every 1/60th of a second, we merely have to count these and increment a clock located at \$F0-\$F2 every second. The timer counter is located at \$EF and is decremented every 1/60th of a second until zero is reached. At that point a timer-finished routine is called.

The clock time is displayed on the video screen at location \$D3F0 every second. This puts it right at the bottom of the CRT screen on the Superboard II and below the scrolling point so that it does not interfere with the BASIC output routines. If you don't have a Superboard II you can put the output wherever your video memory is located. Every second, the time on the clock is compared with the time set in the alarm at location \$F4-\$F6. When the two are equal, an alarm routine is called.

To use this program on the Superboard, you merely have to get into monitor mode, load the program beginning at \$0130, set the alarm for whatever time you want, set the clock for the time of day, load your timer-finished routine and your alarm routine and then flip the switch to turn on the NMI interrupts. Note that the clock and the alarm times are in decimal. They are set with hours in the lowest byte, minutes in the next byte and seconds in the highest byte.

If you don't want the alarm just store a time in it that never occurs in real life such as FF/FF/FF. If you don't want the timer just store a 0 in the timer counter and it will not time out. Note also that you can make this a 24-hour clock by merely changing the instructions at \$0154 and \$0158 to CMPIM \$24 and LDAIM 0, respectively.

Applications

Use the timer to check the position of a joystick for your fancy game program. This will enable the computer to spend the time in between checks computing movements of the ball or spaceship. Also, you might let the timer routine move the ball or tank, etc., on the CRT screen. Then you can change the speed of the game by merely changing the count with which you set the timer. Since the timer counts down in binary it will interrupt after about 4.4 seconds if set to \$FF initially. That would give you time to take aim.

The timer could also be used to switch between two or three

Program listing.

```
                    ;Real Time Clock For The 6502
                    ;By Bruce Hoyt
                    ;
                    ALARM   EQU    $F4
                    TMR     EQU    $EF
                    CLK     EQU    $F0
                    HRS     EQU    $F0
                    SEC     EQU    $F2
                    CYCLE   EQU    $F3
                    VID     EQU    $D3F0
                    USRTMR  EQU    $XXXX
                    USRALM  EQU    $YYYY
                    ;
0130                        ORG    $0130
                    ;
0130  48            TIME    PHA                  ;Save only A for speed
0131  C6F3                  DECZ   CYCLE         ;One cycle count
0133  D04B                  BNE    CHTMR1        ;If ≠ 0 check timer
0135  8A                    TXA                  ;Save X and Y now
0136  48                    PHA                  ;Then update clock
0137  98                    TYA
0138  48                    PHA
0139  F8                    SED                  ;Decimal mode for clock
013A  A93C                  LDAIM  60            ;Reset 60 cycles
013C  85F3                  STAZ   CYCLE
013E  A202                  LDXIM  2
0140  18            INC     CLC                  ;Increment seconds & min
0141  B5F0                  LDAZX  CLK
0143  6901                  ADCIM  1
0145  95F0                  STAZX  CLK
0147  CA                    DEX
0148  300A                  BMI    HRSET
014A  C960                  CMPIM  $60
014C  D00E                  BNE    DISP
014E  A900                  LDAIM  0
0150  95F1                  STAZX  CLK+1
0152  F0EC                  BEQ    INC
0154  C913          HRSET   CMPIM  $13           ;Handle hours separately
0156  D004                  BNE    DISP
0158  A901                  LDAIM  1             ;One O'clock
015A  85F0                  STAZ   HRS
015C  A202          DISP    LDXIM  2             ;Display as HH:MM:SS
015E  A007                  LDYIM  7
0160  B5F0          DISP1   LDAZX  CLK
0162  208D01                JSR    DBYT
0165  A93A                  LDAIM  ':
0167  99F0D3                STAY   VID
016A  88                    DEY
016B  CA                    DEX
016C  10F2                  BPL    DISP1
016E  A202          CHALM   LDXIM  2             ;Check alarm time out
0170  B5F4          CHALM1  LDAZX  ALARM
0172  D5F0                  CMPZX  CLK
0174  D006                  BNE    CHTMR         ;No? then check timer out
0176  CA                    DEX
0177  10F7                  BPL    CHALM1
0179  20YYYY                JSR    USRALM        ;Call your routine
017C  68            CHTMR   PLA
017D  A8                    TAY
017E  68                    PLA
017F  AA                    TAX
0180  A5EF          CHTMR1  LDAZ   TMR           ;If TMR already 0
0182  F007                  BEQ    NOTMR         ;  then no time out
0184  C6EF                  DECZ   TMR
0186  D003                  BNE    NOTMR
0188  20XXXX                JSR    USRTMR        ;Call your routine
018B  68            NOTMR   PLA
018C  40                    RTI
018D  209401        DBYT    JSR    DBYT1         ;Display byte in A
0190  4A                    LSRA                 ;  as two ASCII chars
0191  4A                    LSRA
0192  4A                    LSRA
0193  4A                    LSRA
0194  48            DBYT1   PHA
0195  290F                  ANDIM  $0F
0197  0930                  ORAIM  $30
0199  99F0D3                STAY   VID
019C  88                    DEY
019D  68                    PLA
019E  60                    RTS
```

programs all in memory at the same time . . . as in multitasking. You could put your fancy puzzle solver in the background or your latest PI calculator and run BASIC in the foreground by polling the keyboard every half second or so.

I plan to use the timer to poll the keyboard while I am running some assembly-language programs. That way I can interrupt them when I want to. Hams that have their computer interfaced to their rigs can use the alarm to give automatic station ID every 10 minutes. These are a few of the many uses you may want to try.

In conclusion, I want to point out that though it may seem like this clock routine will take up much of the computer's time, in reality it only uses about .2 percent of the total time! You will never notice the difference unless you are doing some real-time controlling and already driving your 6502 to the limit. But if your 6502 is like mine it spends most of its time waiting! ■

✓Reader Service—see page 258

Wallace Kendall
9002 Dunloggin Rd.
Ellicott City MD 21043

You Can't Hurt It by Pressing the Keys

Treat your computer carefully, but don't deprive inquisitive kids of a chance to use it.

Some day I'm going to find an opportunity to adapt and use a line I heard from my uncle years ago. He was a retired army officer of the World War I era; he hunted, fished, raised bird dogs and, late in life, married and became the father of two attractive and well-behaved children. He had little patience with people who complained about the problems of raising children. "My children and my dogs," he said, "come into the world trained."

My own children were born too long ago to learn about microcomputers in their early years, but I'm keeping my uncle's old line polished and ready. Some day there will be a pause in the brilliant conversation, and though I'll be speaking quietly, everyone in the room will hear me.

"My grandchildren," I'll say, "come into the world knowing BASIC. They learn Pascal before they get into the third grade."

OK, I exaggerate a bit. So did my uncle. But beneath the hyperbole there was a good deal of truth in what my uncle said. His kids could be as wild as the rest, but a quiet word or a lifted eyebrow was all he needed to get them back into line. He never had to wound their dignity with shouts or threats. Come to think of it, he didn't yell at his dogs, either.

Meg, who had not started the first grade when this picture was made, is a Hangman fan. She reads and spells words of seven or eight letters and often guesses words no one thought she would know.

All in the Family

As for those grandchildren I was telling you about: When I got my first computer, a KIM-1, I let them land the moon rocket and hunt the Wumpus and play Hi-Lo.

"You mean you're going to let the kids play with it?" people would say. "Sure," I replied. "You can't hurt it by pressing the keys."

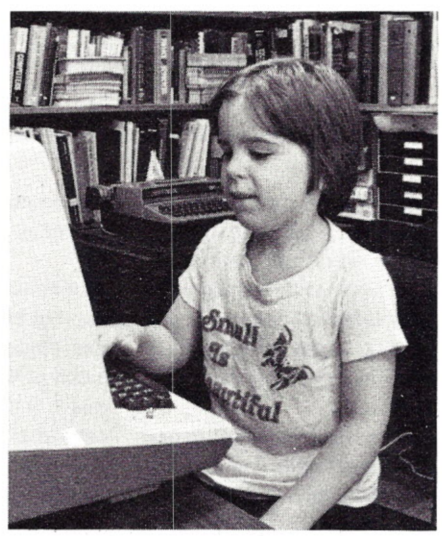
Later I got an OSI Challenger III, with disks and a terminal and a printer. One grandchild is ahead of me in learning BASIC and FORTRAN; he's about 16. One who is six writes letter to her parents using the word processor. One who is three has great fun making lines of different characters using the Repeat key and one character key; he also enjoys playing the beeper with Control G. Others, between three and sixteen, have different degrees of knowledge of BASIC and the operating system, and some are learning a good deal about machine language.

Will they remember tomorrow what they learn today, or are they just playing? There's no way to be sure until tomorrow, I suppose, but I've seen some indications. The youngest was a few months past her second birthday when I visited her family in Louisiana. One evening her father and I turned over his SWTP to her, and I showed her what would happen if she pressed Control G.

Her father moaned, "You've loosed a monster upon us!" Then I understood why he had not already showed her. Until her bedtime we heard "Beep — beep — beeeeeeeeeep." "Oh, well," I said, "she'll probably forget all about it by tomorrow."

After breakfast the next morning she wanted to "play computer." Five seconds after her mother turned it on I heard the "Beeeep."

When the kids started, they were "just playing." The little ones didn't have any idea what any of the keys would do. Now, a few months later, they come to the ter-



Frank, 4, plays Tictactoe, 23 Matches and a few other games. Using the word processor he writes lines of different letters, beeps the beeper, fills the screen and wipes it clean.

minal with a pretty good idea what they want to do. More and more, they are able to do it. One eight-year-old gets annoyed because he wrote a BASIC program that has bugs in it; but he keeps at it, gets help from an older brother, and soon has it running.

Later, the five-year-old is in tears. Her sixteen-year-old brother has been on the terminal too long, and she wants to play Hangman (which calls for spelling words I didn't learn until I was in the fourth grade).

We investigate. He has his school computer on line with the acoustic coupler and doesn't want to give up the terminal until he gets his FORTRAN program compiled. Compilation is soon successful, the computer is changed from one system to another, Hangman is called, and the five-year-old again starts trying to identify and spell words that we consider much too hard for her.

George gets to use the computer fairly often. Here he defers to his cousin Eric from New Mexico.

I will, of course, have to be careful where and when I use the line I intend to steal from my uncle. One thing is certain: I can't use it at my computer club. Some of the members are not over ten or twelve; several of them seem to have come into the world knowing not only BASIC but also theory and hardware.

Occasionally the professionals in the club wander into some pretty high-level engineering discussions. I understand bits and pieces here and there; some of those ten- to sixteen-year-olds seem to understand most of it and to know what kinds of questions to ask about the rest.

It is absolutely infuriating. But I should have known. When you get right down to it, a smart five-year-old can make the same kinds of decisions a computer makes. To the uninitiated it sounds very complicated: "Compare the X register with the Y register: if X is less then Y, increment both X and A; else stop."

A five-year-old, seeing three boxes labeled X, Y and A, can tell whether the Y box has more marbles in it than the X box. He or she can add a marble to the X box and one to the A box, continue until the number in X and Y are equal, and then count the number of marbles in the A box. The computer can do it faster, but not much better for small numbers.

Conclusion

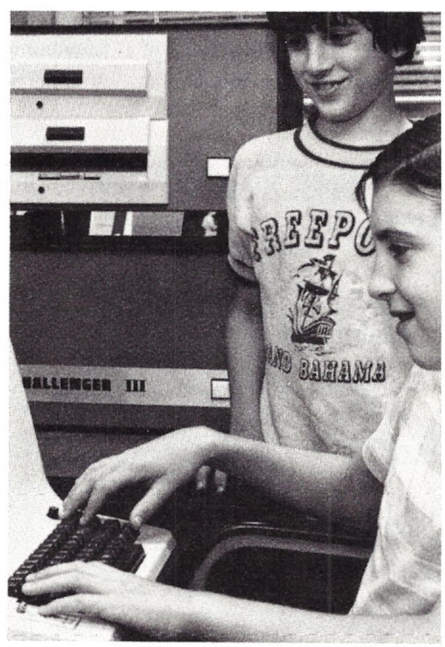
The point is that there are a lot of these kids who are going to change the world faster than those of us who are over 30 can even imagine.

The point is that education starts at birth, not in the first grade.

The point is (now that I think of it) maybe the five-year-old doesn't have to be all that smart to understand a computer. Maybe those ten-year-olds who annoy me at club meetings by understanding so much of the technical talk that goes over my head are not geniuses after all. Maybe they're just average kids who had an opportunity to get their hands on the keys.

But more than anything else, I think, the point is that you can't hurt a computer by pressing the keys. If the two-year-old likes to draw pretty lines with it, why shouldn't he or she? ■

Eight-year-old Cecily likes to write letters on the word processor. Meg keeps an eye on the spelling.



While Christopher tries to corner the last Klingon, Eric waits to do some more work on the BASIC program he wrote to encode and decode messages.

In Search of MWRITE

This circuit detects the absence of a front panel and generates MWRITE accordingly.

Albert S. Woodhull
33 Enfield Rd., RFD 2
Amherst MA 01002

For a school research project that required a small microcomputer system for experimental control, I assembled a few standard S-100 components in a mainframe without a front panel. I had previously developed a monitor program that allowed me to control such a system with a keyboard; on a prototyping card I built a small EPROM memory to hold the monitor. It all worked fine when I tested it in a mainframe with a front panel, but without the front panel it just wouldn't work.

I eventually figured out that many S-100 memory boards required an MWRITE signal, but that some S-100 CPU cards did not provide this signal. In the Imsai equipment I was using, the MWRITE signal came from the front panel. It was easy to generate by using an extra gate in one of the support chips on my EPROM card to perform a NOR operation on the $\overline{\text{PWR}}$ and SOUT lines. However, I wanted to plug my EPROM board into my college's large disk-based system for testing and development. I feared that putting a simple gate on the board could lead to bus conflicts if a front panel did not develop MWRITE in the same way. I needed a circuit that could sense whether a front panel was present on the bus, which could then enable the local MWRITE circuit only when it was needed.

The Circuit Solution

Fig. 1 shows my solution. The front panels on both the Altair and the Imsai systems into which I might want to insert my EPROM board drive the PROT and UNPROT lines on the S-100 bus. If the front panel is present, one of these lines is high and the other is low, but in the absence of a front panel both are pulled up by the inputs on the RAM cards. An XOR gate connected to these lines generates a one-level output if a front panel is present and a 0 level output if the panel is absent. I use the output of this gate to inhibit the MWRITE circuit.

I offer this not just as a solution to a specific problem, but as an illustration of a situation that may come up when modules are switched around between mainframes. With a little extra hardware or software it is possible to make part of a system sensitive to its environment and capable of different modes of operation in differently configured systems. ■

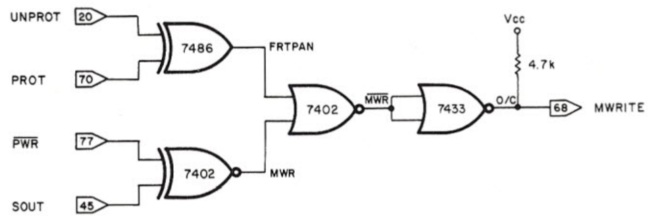


Fig. 1. This circuit allows MWRITE to float if a board (such as a front panel) that develops PROT and UNPROT is present. In many cases, such a board will also normally pull MWRITE low. In a stripped-down system both PROT and UNPROT will float high, and the locally generated MWR is allowed to control the S-100 MWRITE line.

The BASIC Dating Game

You can apply these date-keeping routines to many business applications.

John P. Bauernschub, Jr.
14809 Clavel Street
Rockville MD 20853

This article demonstrates some useful software routines that can be applied to many business applications. Dates are common to all business programs. The easier they are to use, the faster these programs will be written and the more reliable they will be.

The Date Integer

The central idea is to be able to handle the month, day and year in a single variable, called the date integer. Then a few simple routines can be copied into every program to handle all date manipulations. These routines will accept the date from a terminal, check to determine if it is a valid date, then convert it to a unique number. These routines will also reconstruct the month-day-and-year format and print it in an eight-character field.

There are other benefits to using the date-integer concept in addition to programmer productivity and software reliability. Using a single number instead of three numbers for a date saves valuable disk space. Also, dates can now be handled as simple numbers in arithmetic statements.

For example, to check if a customer paid his bill within the discount period, just subtract the number of days in the discount period from the date integer for the day payment was received. Then check if that value is less than the date integer of the billing date. In another example with a payroll program, the pay date can be computed just by adding 7 or 14 to the date integer of the last pay date. This date integer saves the programmer from having to check for month or year changes, or even for leap years.

OK, what is the date integer? It is the total number of days since an arbitrary date. For these routines I selected January 1, 1901, as the starting date. Why? Because the century years 1800, 1900 and 2100 are not leap years, but the year 2000 will be. Years evenly divisible by 100 must also be evenly divisible by 400 to qualify as leap years.

So, by starting in 1901, the program was not required to handle these irregularities, and it still has sufficient latitude for today's business applications. You can easily change the starting date (as will be discussed shortly) to an earlier year and let these century years be leap years.

This looks like another name for the Julian date. True, they both are similar, but the date integer is easier for a programmer to use in arithmetic and comparison routines. This is because he does not have to check if the integer became 78402 or 77963 after adding or subtracting a quantity such as 90 days.

There are two routines in the program listing that convert dates to and from Julian dates using the integer-date routines. They will help you interface existing data files with Julian dates to new programs that use the date integer.

When you use a BASIC interpreter that allows 2-byte integers (you may wish to do this to conserve memory or disk space), the base year should be changed because the maximum positive value for a 2-byte integer is 32,768. This gives a usable span of 89 years. The base year is defined by the variable A5 in line 5030 of the program listing or line 5000 of Program A. The year selected must be a year immediately after a leap year. In such cases, selecting 1961 would be reasonable for most business applications but could not be used for birth dates.

A1	MONTH
A2	DAY
A3	YEAR
A4	COMPUTED DATE INTEGER
A5	BASE YEAR
A6	CONSTANT = 4
A7	CONSTANT = 28
A8	CONSTANT = 29
A9	DATE INTEGER TO BE CONVERTED
X1	DAYS IN A STANDARD YEAR = 365
X2	DAYS IN A LEAP YEAR = 366
X3	DAYS IN FOUR YEARS = 1461
I	FOR-NEXT INDEX VARIABLE
A(12)	DAYS IN EACH MONTH

Table 1. Data dictionary for date-integer routines.

The Program

The program consists of two sections. The first section demonstrates how to use the integer routines. Ten date-manipulation functions are presented in a menu by the program. The program listing of these functions illustrates how to interface with the date-integer subroutines. The code for each function can easily be located by its title in a REM statement. The program can be stopped by entering a zero for the function selection.

The second section begins at line 3000. It has five subroutines that can be inserted into your programs: Initialization; Date Input; Convert Date to an Integer; Convert Integer to Month, Day, Year; and Print Date from an Integer.

Table 1 lists the variables that are used in these five subroutines. These should be reserved and not used for anything else in your application program. Table 2 gives additional variables that are only used in the demonstration section.

The Initialization routine sets the value of constants for the integer conversion routines. This only has to be run once, provided these variables are not changed elsewhere in your program.

The Date-Input-and-Edit routine prompts for the date to be entered in the month, day, year format. These elements are stored in the variables A1, A2 and A3. If the date is not available, enter 0,0,0, and the date integer will be set to zero. This will have significance in the print routine.

The input routine leads into the routine to convert the date to the date integer. These values are checked, and error messages are displayed if it is not a valid date. Then the number of days since the base year is computed, and this value is returned in the variable A4.

The fourth routine converts the date integer in the variable A9 to the month, day and year in variables A1, A2 and A3.

The fifth routine also accepts the date integer in the variable A9. If the value is zero, which means that the date is not available, it prints eight blanks to position the printer where it would have been if there had been a date. When the value in A9 is greater than zero, the subroutine is called to convert the date integer to month, day and year. The date is then printed in the format MM/DD/YY.

It is important to always print eight characters, as mentioned before, to position the print head on a report a fixed distance each time. If your BASIC does not have the PRINT USING command, then line 9100 can be replaced as shown in Example 1 to ensure that the date always occupies eight print positions.

```
9100  PRINT RIGHT$(STR$(A1),2);"/";RIGHT$(STR$(A2),2);"/";
```

Example 1.

It is easy to use the date-integer routines in your application. At the beginning of your program, initialize the constants with a GOSUB 5000 command. When a date is to be entered from the terminal, execute a GOSUB 6000 command. The computer prompts the operator to enter the month, day, year. If the date entered is invalid, a message will be displayed and the date will be requested again. Try entering 6,31,78 or 2,29,78. The subroutine returns the date in A1, A2 and A3 and the date integer in A4. To display a date, let A9 equal the date integer and execute a GOSUB 9000 command.

The date can be entered in either of two formats: MM,DD,YY or MM,DD,YYYY. That is, you can enter the year as 79 or 1979. This way, you can enter dates after the year 2000. If the variable A3 has a value less than 100, then 1900 is added to it. See line 7180.

This program was written in a direct BASIC that should run on all but the simplest BASIC interpreters. For that reason it has only one statement per line. Program A lists the date-integer subroutines in a more compact form for Altair Disk BASIC.

All of the A variables are defined as 2-byte integers with the command DEFINT A. Therefore, the INT function is not used. The reverse slash (\) is for integer division, which is much faster than floating point divi-

D	DATE INTEGER FOR TODAY'S DATE
J	FUNCTION SELECTED
J1	TEMPORARY STORAGE
J2	TEMPORARY STORAGE
J3	TEMPORARY STORAGE
T	TAB VALUE
I$	JULIAN DATE INPUT

Table 2. Data dictionary for demonstration routines.

```
5000 DEFINTA:DIMA(12):A5=1901:A6=4:A7=28:A8=29:X1=365:X2=366:X3=1461
5010 FOR I=1TO12:READ A(I):NEXT:RETURN
5020 DATA 31,29,31,30,31,30,31,31,30,31,30,31
6000 INPUT"DATE";A1,A2,A3
7000 IFA1=0XORA2=0THENPRINT"BOTH MONTH & DAY MUST BE 0 OR NEITHER CAN BE
0... REENTER ";:GOTO6000
7010 IFA1<0ORA1>12THENPRINT"MONTH ERROR... REENTER ";:GOTO6000
7020 IFA2<0ORA2>A(A1)THENPRINT"DAY ERROR... REENTER ";:A(2)=A8:GOTO6000
7030 IFA1=0ANDA2=0THENA3=0:A4=0:RETURN
7040 IFA3<100THENA3=A3+1900
7050 IFA3<A5ORA3>1989THENPRINT"THE YEAR IS INVALID.  REENTER ";:GOTO6000
7060 A4=(A3-A5)*X1+(A3-A5)\A6
7070 IFA3-(A3\A6)*A6=0THENA(2)=A8ELSEA(2)=A7:IFA2>A(A1)GOTO7020
7080 IFA1=1GOTO7100
7090 FORI=1TOA1-1:A4=A4+A(I):NEXT
7100 A4=A4+A2:A(2)=A8:RETURN
8000 A3=(A9-1)\X3:A9=A9-A3*X3:A3=A3*A6+A5:A(2)=A7:I=0:A1=0
8010 IFA9-X1<1GOTO8030
8020 A9=A9-X1:A3=A3+1:I=I+1:IFI<3GOTO8010ELSEA(2)=A8
8030 A1=A1+1:IFA9-A(A1)<1GOTO8040ELSEA9=A9-A(A1):GOTO8030
8040 A2=A9:A(2)=A8:RETURN
9000 IFA9>0THENGOSUB8000ELSEPRINT"        ";:RETURN
9010 PRINTUSING"##/";A1;A2;:PRINTRIGHT$(STR$(A3),2);:RETURN
```

Program A. Date-integer routines in compact format.

Program listing.

```
1000 REM          T H E   B A S I C   D A T I N G   G A M E
1010 REM
1020 REM WRITTEN BY:  JOHN P. BAUERNSCHUB, JR.
1030 REM OCTOBER 25, 1978
1040 REM
1050 GOSUB 5000
1060 PRINT CHR$(26)
1070 LET T=20
1080 PRINT TAB(T);"ENTER TODAY'S ";
```

```
1090 GOSUB 6000
1100 LET D=A4
1110 PRINT
1120 PRINT
1130 IF J>0 GOTO 1240
1140 PRINT TAB(T);"1 - CONVERT DATE TO AN INTEGER"
1150 PRINT TAB(T);"2 - CONVERT AN INTEGER TO A DATE"
1160 PRINT TAB(T);"3 - DAY NUMBER IN THIS YEAR"
1170 PRINT TAB(T);"4 - DAYS REMAINING THIS YEAR"
1180 PRINT TAB(T);"5 - DATE N DAYS FROM TODAY (+ OR -)"
1190 PRINT TAB(T);"6 - ADD (+ OR -) DAYS TO ANY DATE"
1200 PRINT TAB(T);"7 - NUMBER OF DAYS BETWEEN TWO DATES"
1210 PRINT TAB(T);"8 - DAY IN YOUR LIFE"
1220 PRINT TAB(T);"9 - CONVERT DATE TO JULIAN DATE"
1230 PRINT TAB(T-1);"10 - CONVERT JULIAN DATE TO DATE"
1240 PRINT
1250 PRINT TAB(T);"    SELECT A FUNCTION";
1260 INPUT J
1270 PRINT
1280 IF J=0 THEN STOP
1290 ON J GOSUB 1320,1390,1450,1550,1650,1700,1800,1950,2020,2150
1300 GOTO 1110
1310 REM
1320 REM   CONVERT DATE TO AN INTEGER
1330 REM
1340 PRINT "ENTER THE ";
1350 GOSUB 6000
1360 PRINT A4
1370 RETURN
1380 REM
1390 REM   CONVERT INTEGER TO A DATE
1400 REM
1410 INPUT "ENTER THE INTEGER";A9
1420 GOSUB 9000
1430 RETURN
1440 REM
1450 REM   DAY NUMBER IN THIS YEAR
1460 REM
1470 LET A9=D
1480 GOSUB 8000
1490 LET A1=1
1500 LET A2=1
1510 GOSUB 7000
1520 PRINT "TODAY IS DAY NUMBER";D-A4+1;"IN";A3
1530 RETURN
1540 REM
1550 REM   DAYS REMAINING THIS YEAR
1560 REM
1570 LET A9=D
1580 GOSUB 8000
1590 LET A1=12
1600 LET A2=31
1610 GOSUB 7000
1620 PRINT "THERE ARE";A4-D;"DAYS REMAINING THIS YEAR"
1630 RETURN
1640 REM
1650 REM   COMPUTE A DATE FROM TODAY
1660 REM
1670 LET J1=D
1680 GOTO 1750
1690 REM
1700 REM   COMPUTE A DATE FROM ANY DATE
1710 REM
1720 PRINT "ENTER THE ";
1730 GOSUB 6000
1740 LET J1=A4
1750 INPUT "ENTER NUMBER OF DAYS TO BE ADDED (+ OR -)";J2
1760 LET A9=J1+J2
1770 GOSUB 9000
1780 RETURN
1790 REM
1800 REM   NUMBER OF DAYS BETWEEN TWO DATES
1810 REM
1820 PRINT "ENTER THE EARLIER ";
1830 GOSUB 6000
1840 LET J1=A4
1850 PRINT "ENTER THE LATER ";
1860 GOSUB 6000
1870 PRINT "THERE ARE";A4-J1;"DAYS BETWEEN ";
1880 LET A9=J1
1890 GOSUB 9000
1900 PRINT " AND ";
1910 LET A9=A4
1920 GOSUB 9000
1930 RETURN
1940 REM
1950 REM   COMPUTE THE DAY IN YOUR LIFE
1960 REM
1970 PRINT "ENTER YOUR BIRTH";
1980 GOSUB 6000
1990 PRINT "THIS IS DAY NUMBER";D-A4+1;"IN YOUR LIFE"
2000 RETURN
2010 REM
2020 REM   CONVERT DATE TO JULIAN DATE
2030 REM
2040 PRINT "ENTER THE ";
2050 GOSUB 6000
2060 LET J1=VAL(RIGHT$(STR$(A3),2))*1000
2070 LET J3=A4
2080 LET A1=1
2090 LET A2=1
2100 GOSUB 7000
2110 LET J2=J1+J3-A4+1
2120 PRINT J2
2130 RETURN
2140 REM
2150 REM   CONVERT JULIAN DATE TO DATE
2160 REM
2170 INPUT "ENTER THE JULIAN DATE (YYDDD)";I$
2180 LET J1=INT(VAL(I$)/1000)+1900
2190 IF J1=>A5 THEN GOTO 2220
2200 PRINT "THE YEAR CANNOT BE BEFORE ";A5
2210 GOTO 2150
2220 LET A1=1
2230 LET A2=1
2240 LET A3=J1
2250 GOSUB 7000
2260 LET J2=VAL(RIGHT$(I$,3))
2270 IF J2<1 THEN GOTO 2310
2280 LET J3=X1
2290 IF J1/4=INT(J1/4) THEN J3=X2
2300 IF J2<=J3 THEN GOTO 2330
2310 PRINT "DAY ERROR... RE";
2320 GOTO 2150
2330 A9=A4+J2-1
2340 GOSUB 9000
2350 RETURN
2360 REM
2370 REM
3000 REM   * * * * * * * * * * * * * * * * * * *
3010 REM
3020 REM         D A T E   I N T E G E R   R O U T I N E S
3030 REM
3040 REM   * * * * * * * * * * * * * * * * * * *
3050 REM
5000 REM         I N I T I A L I Z A T I O N
5010 REM
5020 DIM A(12)
5030 LET A5=1901
5040 LET A6=4
5050 LET A7=28
5060 LET A8=29
5070 LET X1=365
5080 LET X2=366
5090 LET X3=1461
5100 FOR I=1 TO 12
5110 READ A(I)
5120 NEXT I
5130 DATA 31,29,31,30,31,30,31,31,30,31,30,31
5140 RETURN
5150 REM
6000 REM         D A T E   I N P U T
6010 REM
6020 INPUT "DATE (M,D,Y)";A1,A2,A3
6030 REM
7000 REM         C O N V E R T   D A T E   T O   I N T E G E R
```

```
7010 REM
7020 REM     INPUT DATA IS IN A1, A2, A3
7030 REM     RETURNS DATE INTEGER IN A4
7040 REM
7050 IF A1<0 THEN GOTO 7310
7060 IF A1>12 THEN GOTO 7310
7070 IF A2<0 THEN GOTO 7330
7080 IF A1>0 THEN GOTO 7130
7090 IF A2>0 THEN GOTO 7140
7100 LET A3=A5
7110 LET A4=0
7120 GOTO 7290
7130 IF A2>0 THEN GOTO 7170
7140 PRINT "BOTH MONTH & DAY MUST BE ZERO OR NEITHER CAN BE ZERO..."
7150 PRINT "REENTER ";
7160 GOTO 6020
7170 IF A2>A(A1) THEN GOTO 7330
7180 IF A3<100 THEN LET A3=A3+1900
7190 IF A3<A5 THEN GOTO 7360
7200 LET A4=(A3-A5)*X1+INT((A3-A5)/A6)
7210 LET A(2)=A7
7220 IF A3-INT(A3/A6)*A6=0 THEN LET A(2)=A8
7230 IF A2>A(A1) THEN GOTO 7330
7240 IF A1=1 THEN GOTO 7280
7250 FOR I=1 TO A1-1
7260 LET A4=A4+A(I)
7270 NEXT I
7280 LET A4=A4+A2
7290 LET A(2)=A8
7300 RETURN
7310 PRINT "MONTH ERROR... REENTER ";
7320 GOTO 6020
7330 PRINT "DAY ERROR... REENTER ";
7340 LET A(2)=A8
7350 GOTO 6020
7360 PRINT "THE YEAR CANNOT BE BEFORE ";A5;"  REENTER ";
7370 GOTO 6020
7380 REM
8000 REM     C O N V E R T   I N T E G E R   T O
8010 REM       M O N T H ,   D A Y   A N D   Y E A R
8020 REM
8030 REM     INPUT DATA IS IN A9
8040 REM     RETURNS A1, A2, A3
8050 REM
8060 LET A3=INT((A9-1)/X3)
8070 LET A9=A9-A3*X3
8080 LET A3=A3*A6+A5
8090 LET A(2)=A7
8100 LET I=0
8110 LET A1=0
8120 IF A9-X1<1 THEN GOTO 8180
8130 LET A9=A9-X1
8140 LET A3=A3+1
8150 LET I=I+1
8160 IF I<3 THEN GOTO 8120
8170 LET A(2)=A8
8180 LET A1=A1+1
8190 IF A9-A(A1)<1 THEN GOTO 8220
8200 LET A9=A9-A(A1)
8210 GOTO 8180
8220 LET A2=A9
8230 LET A(2)=A8
8240 RETURN
8250 REM
9000 REM     P R I N T   D A T E   F R O M   I N T E G E R
9010 REM
9020 REM     DATA IS IN A9
9030 REM
9040 IF A9=0 THEN PRINT "        ";
9050 IF A9=0 THEN RETURN
9060 GOSUB 8000
9070 REM
9080 REM     P R I N T   D A T E   A S   MM/DD/YY
9090 REM
9100 PRINT USING "##/";A1;A2;
9110 PRINT RIGHT$(STR$(A3),2);
9120 RETURN
```

sion. This code should also work on Radio Shack Level II BASIC. Program A can be used with the demonstration routines without any changes since the same variables and subroutine entry points are used.

Dates can be awkward to use. However, with a little standardization, a programmer can handle them with confidence. ■

Sample run.

```
ENTER TODAY'S DATE (M,D,Y)? 10,25,78

 1 - CONVERT DATE TO AN INTEGER
 2 - CONVERT AN INTEGER TO A DATE
 3 - DAY NUMBER IN THIS YEAR
 4 - DAYS REMAINING THIS YEAR
 5 - DATE N DAYS FROM TODAY (+ OR -)
 6 - ADD (+ OR -) DAYS TO ANY DATE
 7 - NUMBER OF DAYS BETWEEN TWO DATES
 8 - DAY IN YOUR LIFE
 9 - CONVERT DATE TO JULIAN DATE
10 - CONVERT JULIAN DATE TO DATE

     SELECT A FUNCTION? 1

ENTER THE DATE (M,D,Y)? 10,25,78
 28422

     SELECT A FUNCTION? 2

ENTER THE INTEGER? 28422
10/25/78

     SELECT A FUNCTION? 3

TODAY IS DAY NUMBER 298 IN 1978

     SELECT A FUNCTION? 4

THERE ARE 67 DAYS REMAINING THIS YEAR

     SELECT A FUNCTION? 5

ENTER NUMBER OF DAYS TO BE ADDED (+ OR -)? 45
12/ 9/78

     SELECT A FUNCTION? 6

ENTER THE DATE (M,D,Y)? 12,10,78
ENTER NUMBER OF DAYS TO BE ADDED (+ OR -)? 70
 2/18/79

     SELECT A FUNCTION? 7

ENTER THE EARLIER DATE (M,D,Y)? 4,15,78
ENTER THE LATER DATE (M,D,Y)? 5,6,78
THERE ARE 21 DAYS BETWEEN  4/15/78 AND  5/ 6/78

     SELECT A FUNCTION? 8

ENTER YOUR BIRTHDATE (M,D,Y)? 5,8,1953
THIS IS DAY NUMBER 9302 IN YOUR LIFE

     SELECT A FUNCTION? 9

ENTER THE DATE (M,D,Y)? 10,25,78
 78298

     SELECT A FUNCTION? 10

ENTER THE JULIAN DATE (YYDDD)? 80060
 2/29/80

     SELECT A FUNCTION? 0

BREAK IN 1270
OK
```

DYNACOMP

Quality software for: Apple II Plus
TRS-80 (Level II)
North Star

All software is supplied with complete documentation which includes clear explanations and examples. Each program will run with standard terminals (32 characters or wider) and within 16K program memory space. Except where noted, all software is available on North Star diskette (North Star BASIC), TRS-80 cassette (Level II) and Apple cassette (*Applesoft* BASIC). These programs are also available on *PAPER TAPE* (Microsoft *BASIC*).

FLIGHT SIMULATOR — Price: $17.95 postpaid
(as described in SIMULATION, Volume II)
A realistic and extensive mathematical simulation of take-off, flight and landing. The program utilizes aerodynamic equations and the characteristics of a real airfoil. You can practice instrument approaches and navigation using radials and compass headings. The more advanced flyer can also perform loops, half-rolls and similar aerobatic maneuvers.

SIMULATION, Volume II (BYTE Publications): $6.00

VALDEZ — Price: $14.95 postpaid
A simulation of supertanker navigation in the Prince William Sound and Valdez Narrows. The program uses an extensive 256X256 element radar map and employs physical models of ship response and tidal patterns. Chart your own course through ship and iceberg traffic. Any standard terminal may be used for display.

BRIDGE 2.0 — Price: $17.95 postpaid
An all-inclusive version of this most popular of card games. This program both BIDS and PLAYS either contract or duplicate bridge. Depending on the contract, your computer opponents will either play the offense OR defense. If you bid too high the computer will double your contract! BRIDGE 2.0 provides challenging entertainment for advanced players and is an excellent learning tool for the bridge novice.

HEARTS 1.5 — Price: $14.95 postpaid
An exciting and entertaining computer version of this popular card game. Hearts is a trick-oriented game in which the purpose is not to take any hearts or the queen of spades. Play against two computer opponents who are armed with hard-to-beat playing strategies.

DATA SMOOTHER — Price: $14.95 postpaid
This special data smoothing program may be used to rapidly derive useful information from noisy business and engineering data which are equally spaced. The software features choice in degree and range of fit, as well as smoothed first and second derivative calculation. Also included is automatic plotting of the input data and smoothed results.

FOURIER ANALYZER — Price: $14.95 postpaid
Use this program to examine the frequency spectra of limited duration signals. The program features automatic scaling and plotting of the input data and results. Practical applications include the analysis of complicated patterns in such fields as electronics, communications and business.

MAIL LIST I — Price: $18.95 postpaid (available for North Star only)
A many-featured mailing list program which searches through your customer list by user-defined product code, customer name or Zip Code. Entries to the list can be conveniently added or deleted and the printout format allows the use of standard size address labels. Each diskette can store more than 1000 entries.

MAIL LIST SERVICE
DYNACOMP can provide you with a customized mail list service. Your customer/patient records are placed in a master computer file and you are provided with addressed, self-adhesive labels for your mailings. These labels may be sorted by name, Zip Code, date, or other identifiers. Write for further details and a price schedule.

TEXT EDITOR I (Letter Writer) — Price: $14.95 postpaid
An easy to use, line-oriented text editor which provides variable line widths and simple paragraph indexing. This text editor is ideally suited for composing letters and is quite capable of handling much larger jobs.

GAMES PACK I — Price: $10.95 postpaid
Seven entertaining games for less than a dollar a kilobyte! Play CATAPULT, CRAPS, SWITCH, HORSERACE, SLOT MACHINE, BLACKJACK and LUNAR LANDER. This is an excellent and economical way to start your games library.

All orders are processed within 48 hours. Please enclose payment with order. If paying by MASTER CHARGE or VISA, include all numbers on card. Foreign orders add 10% for shipping and handling.

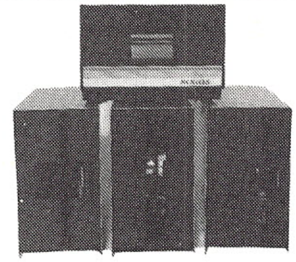
Write for detailed descriptions of these and other programs available from **DYNACOMP**.

VISA — master charge

DYNACOMP ✓161
P.O. Box 162 Dept. K
Webster, New York, 14580
New York State residents please add 7% NYS sales tax.

SAVE ON ADD-ON PRODUCTS FOR TRS-80

The largest family of disk drives from the largest supplier, drives come complete with power supply and cabinet.



MTI-40 Disk Drive, 35 & 40 track	$369
TF-1 Perfec FD200, 40 track, use both sides	$389
TF-3 Shugart SA400, 35 track, same as tandy	$389
TF-5 MPI B51, 40 track	$379
TF-70 Micropolis, 77 track with 195K of storage	$639
TDH-1 Dual sided drive, 35 track	$499
Maxi Disk 2: 10 Megabyte (fixed) Winchester Technology	$5349

NEW PRODUCTS

16K Memory	$86
Modem	$179
Expansion Interface 32K	$499
AC Isolator	$47.95

PRINTERS

DP800 Anadex, 80 column, 112cps	$949
LP779 Centronics 779	$1099
LP730 Centronics 730	$950
LP700 Centronics 700	$1395
LP701 Centronics 701	$1759
LP702 Centronics 702	$1995
SPW-1 Spinwriter-NEC	$2525

NEW! LINE PRINTER BASE 2

Base 2 Printer 80, 132 col., graphics 60 LPM with tractors **$599**

*** DRIVES FOR ANY MICROCOMPUTER ***
Does not include power supply & cabinet.

MOD II DISK DRIVES NOW AVAILABLE

Perfec FD200	$282
Perfec FD250 (dual head)	$399
Shugart SA400 (unused)	$286
Shugart SA800	$479
MPI B52 .... $349 B51	$279

SOFTWARE

Disk Drive Motor Speed Test	$19.95
New DOS+ with over 200 modifications and corrections to TRS-DOS	$99
New DOS+ 40 track	$110
AJA Word Processor	$75
AJA Business Program	$250
Racet Infinite Basic	$49.95
Disk Drive Alignment Program	$109
Radix Data Base Program	$99.95
Electric Pencil	$150

ALL PRICES CASH DISCOUNTED. FREIGHT FOB/FACTORY

MICROCOMPUTER TECHNOLOGY INCORPORATED
3304 W. MacArthur
Santa Ana, CA 92704
(714) 979-9923

Apparat, Inc. ✓30
7310 E. Princeton Ave.
Denver, CO 80222
(303) 758-7275

Gary Sabot
38 Woodland Road
Roslyn NY 11576

You Name It!

If you're one of those people who have trouble with memory (remembering names), here's a program that allows you to devise your own five-letter names for things.

```
1 REM **NAME IT!** BY GARY SABOT
10 REM ONLY 20 CONSONANTS ARE USED SINCE Q IS OMITTED
30 RANDOM
40 DIM C(3), V(3)
50 REM LINE 51 CAUSES A NEW PAGE TO BE GENERATED
51 PRINT CHR(5);
60 LET A$="AEIOU"
70 LET B$="BCDFGHJKLMNPRSTVWXYZ"
80 LET C=1
90 PRINT "HOW MANY RANDOM WORDS DO YOU WANT";
100 INPUT G
110 FOR B=1 TO G
120 REM RANDOM ROUTINE TO DETERMINE WHICH
130 REM CHARACTERS WILL BE PRINTED
140 FOR K=1 TO 3
150 LET V(K)=INT(5*RND(0))+1
160 LET C(K)=INT(20*RND(0))+1
170 NEXT K
180 REM WORD PRINTING ROUTINE
190 PRINT MID(B$,C(1),1);MID(A$,V(1),1);
200 PRINT MID(B$,C(2),1);MID(A$,V(2),1);
210 PRINT MID(B$,C(3),1);"     ";
220 REM ROUTINE TO FORMAT PRINTOUT TO 7 WORDS/LINE
230 LET C=C+1
240 IF C>7 THEN 260
250 GOTO 280
260 PRINT
270 LET C=1
280 NEXT B
290 END
```

Fig. 1. Program listing.

The following program arose from the need to name a parakeet I had just purchased. I had heard about this sort of program before, and since I couldn't find an actual copy of it, I decided to write it myself.

A Basic Explanation

This program is designed to print out five-letter words. The first, third and fifth letters are consonants, while the second and forth are vowels.

Lines 30 through 80 are used to initialize variables and to store the vowels and consonants in strings A$ and B$, respectively. I only used 20 consonants, leaving the q out. This was because, in English, q is almost always followed by u, while in my program it quite often wasn't, and it caused many unpronounceable combinations.

Lines 90 and 100 are used

to determine how many words the user wants. Lines 110 through 290 form the main loop of the program, which is repeated once for each word to be printed.

Line 150 is used to fill an array, V, which is dimensioned to 3, with random numbers from 1 to 5. These numbers will later be used to determine which of the five vowels to print in the two spaces allotted for them. One of the spots in the array is not used in the printing. Line 160 fills a similar array, C, with random numbers from 1 to 20 for each of the 20 consonants.

Lines 190 through 210 print the word that has been computed with the random numbers. In the BASIC that I was using, the MID statement, which is employed in the printing routine, is used as follows: The statement MID(Z$,A,B) would mean print a substring of Z$, starting at character A, and for a length of B characters. After the word has been printed, four spaces are printed to separate the words.

Lines 230 through 280 are used to format the words into seven words per line. This ensures that in a standard 64-character line, none of the words will be chopped in half, leaving two letters on one line and three on another, for example. If your computer has a different line length, just change the 7 in line 240 to the number of words that your computer can fit on a line, and the program will take care of the rest. Fig. 1 shows the program, and Fig. 2 shows a sample run.

By the way, my parakeet's name is "Holmes" (I added an I to the word "homes," which is what the computer actually provided). ■

```
RUN

HOW MANY WORDS DO YOU WANT? 35

VOLEG   GAZEL   TENAN   ROZAR   ZANEZ   TOJEK   BAHEN

WETAL   RUXEZ   CABUH   GEJEJ   POZIS   LODAR   SILEL

HEJEH   RIBUY   SOMUG   RULOB   CIFUW   VOTIX   WILEV

YUZOZ   DELOW   ZUCOP   FAMUV   REJEY   ZUTUH   RUSAF

JEBUY   GAWER   KAHUX   BEWAP   NARAL   VUYON   LADEL
```

Fig. 2. Sample run.

Reader Service—see page 258

Cut 'Em off at the Pass!

Looking for something light in Level I? You'll like this a lot.

Richard A. Rodman
1727 Cy Court
Vienna VA 22180

Here's a fun game in Level I BASIC that demonstrates the graphics capabilities of the TRS-80. It could be modified to run on almost any machine with graphics. Two can play; the computer is a passive participant.

Each player has a dot that moves on the screen leaving a trail behind it. He must avoid striking or passing over any of the old trails. The object is to cut off the other player so he cannot find any open space.

Each player, on his turn, types U, D, L or R to indicate in which direction he wants to move. Then, his dot will move a random distance up to ten spaces. The upper line of the display is saved for user prompts; the remainder of the screen is the playing field. If you go off any side, you'll reappear on the opposite side.

Astute video-game fanatics will recognize that this game is similar to the Atari game "Surround" and the G. I. Gimini game "Barricade." However, the nonreal-time operation in BASIC makes a great difference.

The game is quite simple and, since it's in BASIC, lends itself to easy modification. It's only about 1132 bytes long. One possible modification would be the addition of extra blocks, walls or other obstacles . . . or add secret passageways that cause players to disappear and reappear somewhere else!

Notice the statements around line 1000. These add/subtract M to/from X/Y. If M = 1, the trails are solid. Making M more than one will give a "spaced out" effect.

All of my friends who played this game liked it a lot! ■

CLS:	Clear screen.
R.(N) or RND(N):	Random function; if N>0, returns an integer from 1 to N.
SET(X,Y):	Turn on the bit at X,Y (make it white).
RESET(X,Y):	Turn off the bit at X,Y (make it black).
POINT(X,Y):	Returns 1 if X,Y is on, 0 if it is off.
P.:	Print.
P.AT 0:	Print on the first line.
G.:	GOTO.
GOS.:	GOSUB.
F.:	FOR.
T.:	THEN.
IN.:	INPUT.
N.:	NEXT.
RET.:	RETURN.

Table 1. Abbreviations have been used in this program. These abbreviations are particular to the TRS-80, so an explanation of what they stand for is included for Poly 88 users and others who may want to use this program.

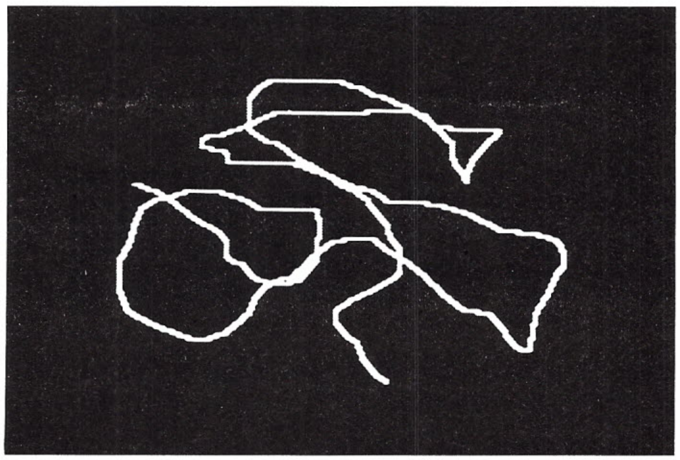
```
1 CLS:P."CUT 'EM OFF AT THE PASS!"
2 P.:P."TO GO:  ↑  ↓  ←  →"
3 P.   "TYPE:   U  D  L  R"
4 P.:P."WHEN I TELL YOU IT'S YOUR TURN, TYPE ONE OF THOSE FOUR"
5 P."LETTERS. SORRY, NO TYPING ERRORS ALLOWED!"
6 P.:P."THE OBJECT OF THIS GAME IS TO AVOID RUNNING INTO ANY OF THE"
7 P."OLD TRACKS.  YOU'LL SEE WHAT I MEAN!"
8 P.
10 IN."WHO IS THE PLAYER ON THE LEFT";A$
15 IN."AND WHO IS ON THE RIGHT";B$
20 U=1:D=2:R=3:L=4:Q=0:CLS
25 M=1:REM MOVE INCREMENT
30 N=10/M:REM MOVE LIMIT
35 A=30:I=25:C=90:G=25:SET(A,I):SET(C,G)
100 REM FIRST PLAYER GOES
110 P.AT 0,A$;:IN.E:H=RND(N)
120 X=A:Y=I:GOS.1000
125 A=X:I=Y
130 IF Q=0 G. 210
140 P.AT 0,"SORRY,";A$:G.10
200 REM SECOND PLAYER GOES
210 P.AT 0,B$;:IN.E:H=RND(N)
220 X=C:Y=G:GOS.1000
225 C=X:G=Y
230 IF Q=0 G. 110
240 P.AT 0,"SORRY,";B$:G. 10
1000 F.T=1 TO H:REM THIS ROUTINE MOVES THE DOT
1005 ON E G. 1010,1020,1030,1040
1010 Y=Y-M:G. 1050
1020 Y=Y+M:G. 1050
1030 X=X+M:G. 1050
1040 X=X-M
1050 IF X<0 T. X=X+128:REM NORMALIZE
1060 IF Y<3 T. Y=Y+45
1070 IF X>127 T. X=X-128
1080 IF Y>47 T. Y=Y-45
1100 Q=Q+POINT(X,Y):SET(X,Y)
1110 N.T
1115 P.AT 0,"                         ";
1120 RET.
1130 END
```

Program listing.

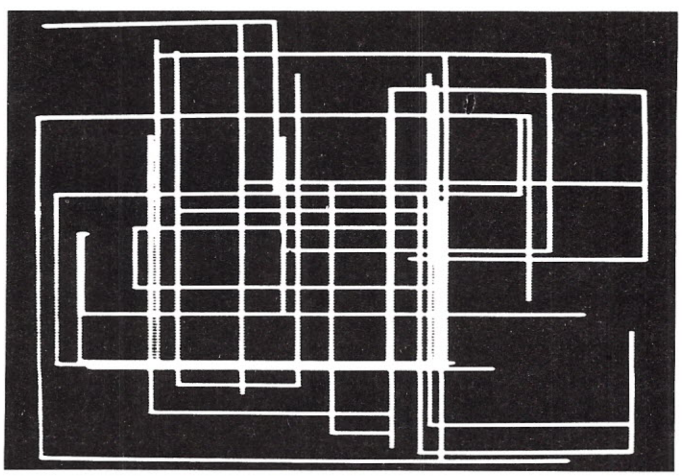
✓Reader Service—see page 258

An Easy Way to Apple Hi-Res Graphics

Assembly language is not needed for high-resolution graphics on the Apple II.



Example of Hi-Res Sketch subprogram using Apple II's game paddles.



Example of Random Verticals and Horizontals program.

```
8199 REM . . .'GOSUB WIPE' TO CLEAR SCREEN BY PUSHING PADDLE BUTTON.
8200 IF PEEK (-16287)>127 OR PEEK (-16286)>127 THEN CALL CLEAR
```

Fig. 1.

Malcolm J. R. Clark
336 Foul Bay Road
Victoria, British Columbia
Canada V8S 4G7

As soon as my 16K Apple II arrived, I was rather keen to get going with the high-resolution graphics, but I soon discovered that I would need to learn to program in assembler. However, the Hi-Res demo cassette the Apple people supplied had a machine-language program all ready to go, followed by a BASIC program that interacted with the first program. Therefore, what could be easier than loading only the machine-language program, followed by a different BASIC program of one's own devising within the limited unused memory available?

PDL(0)	PDL(1)
119	15
111	50
96	33
80	71
69	59
65	77
65	76
65	77
65	77
65	76
65	78
65	77
66	72
70	86
73	71
77	55
80	72
84	54
86	46
87	41

Table 1. Example printout for statements "2 PRINT PDL(0), PDL1: GOTO 2." Only paddle 0 was manipulated.

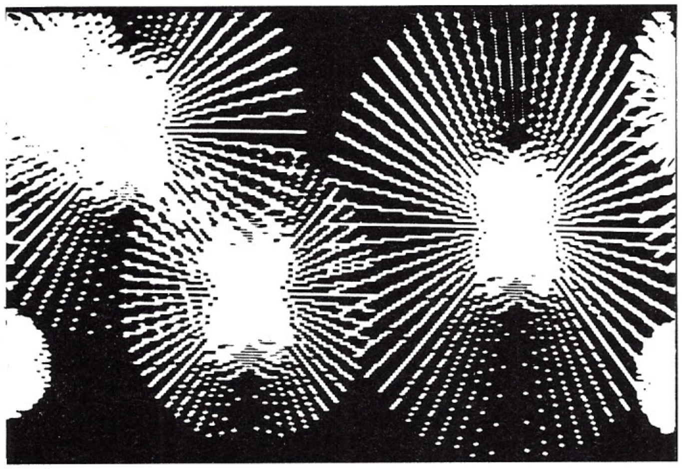
The program listing shows the substitute BASIC program I prepared to interact with the Hi-Res machine language. This program contains four subprograms: one to allow free-hand drawing with the game paddles, one to draw random horizontal and vertical lines, one to sketch the starship Enterprise with zoom and, last but best, Clark's Art Gallery. All four take advantage of the fun involved in watching the computer draw a shape a line at a time as opposed to the opposite strategy of flashing up a drawing apparently instantaneously. Please keep in mind that the program assumes the game paddles are plugged in, the machine-language program loaded and HIMEM set. The accompanying photographs show several example plots.

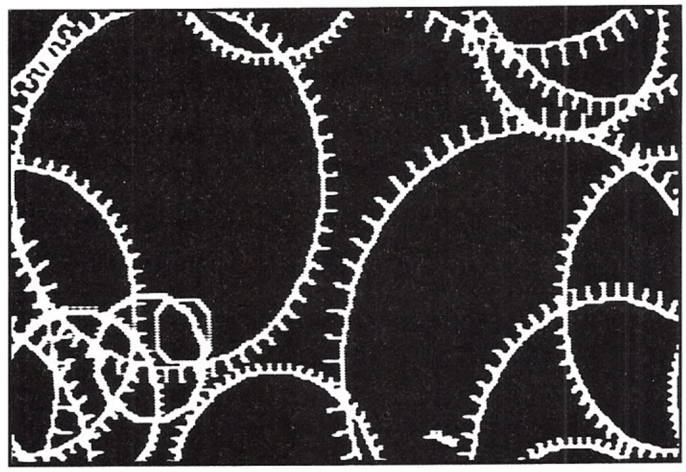
The Sketch subprogram surprised me with occasional erratic results, and it turned out that the two game paddles were not independent. To illustrate this, key and run the program

```
2 PRINT PDL(0), PDL(1): GOTO 2
```

and alternately change the settings of one paddle, then the other. Table 1 shows an example of PDL(1) values changing, though only PDL(0) was manipulated. For the two Apples I've tried this on, it seems worse for high settings of PDL(1).

You'll see another surprise if you take a look at statement





Examples from Clark's Art Gallery.

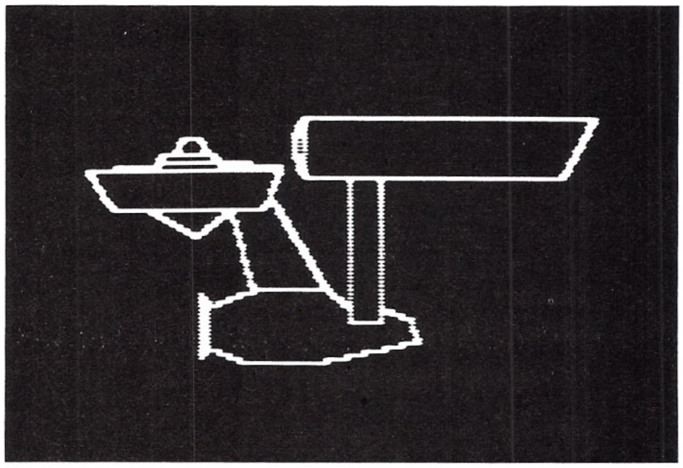
Program listing.

```
10 REM :...SOME EXAMPLES OF HIGH RESOLUTION GRAPHICS PROGRAMS IN AP
   PLE II INTEGER BASIC"
20 REM  ....M. J. R. CLARK......1978
30 REM : LOAD THE MACHINE LANGUAGE PROGRAM FROM APPLE'S HIGH-RES DE
   MO TAPE BEFORE LOADING THIS.
40 REM *****REMEMBER TO SET HIMEM ****
100 INIT=3072:CLEAR=3086:POSN=3761:PLOT=3780:LINE=3786:POINT=8000
   :FIRST=8100:WIPE=8200:KEY=8300:TELL=8360
101 SHAPE=3805
110 DIM E(56),C(4),Y$(1)
130 W=255:C(1)=W: REM ...WHITE
140 V=170:C(2)=V: REM ...VIOLET
150 G=85:C(3)=G: REM ...GREEN
160 B=0:C(4)=B: REM ...BLACK
1000 POKE -16303,0: CALL -936
1010 PRINT "........HIGH RESOLUTION GRAPHICS........": PRINT
1020 PRINT "PROGRAMS IN APPLE II INTEGER BASIC"
1030 PRINT : PRINT "MACHINE LANGUAGE PROGRAM FROM APPLE'S   HIGH-RESO
   LUTION DEMO TAPE MUST BE LOADED PRIOR TO THIS PROGRAM."
1035 PRINT "HIMEM ALSO MUST HAVE BEEN SET"
1036 PRINT : PRINT
1040 PRINT : PRINT "THE FOLLOWING EXAMPLE PROGRAMS ARE READY TO RUN:
   ": PRINT
1050 PRINT " 1...HI-RES SKETCH USING GAME PADDLES"
1060 PRINT " 2...RANDOM VERTICALS & HORIZONTALS"
1070 PRINT " 3...SIMPLE 2-D WITH ZOOM"
1080 PRINT " 4...CLARK'S ART GALLERY"
1200 PRINT "99...END JOB"
1300 PRINT : PRINT : PRINT
1400 INPUT "WHICH ONE DO YOU WISH",WHICH
1410 IF WHICH>0 AND WHICH<=4 THEN GOTO 1440
1415 IF WHICH=99 THEN END
1420 PRINT "ERROR FOUND--PLEASE TRY AGAIN"
1430 GOTO 1400
1440 GOTO 1000+500*WHICH
1450 END
1500 REM ...HIGH RESOLUTION SKETCH
1510 CALL INIT
1520 GOSUB FIRST
1525 GOSUB TELL
1530 X= PDL (0)
1540 Y= PDL (1)
1550 IF Y>159 THEN Y=159
1560 GOSUB WIPE
1570 GOSUB KEY
1580 COLOUR=W
1590 GOSUB POINT
1600 CALL LINE
1610 GOTO 1530
2000 REM ...RANDOM VERTICALS AND HORIZONTALS
2010 CALL INIT
2020 GOSUB FIRST
2025 GOSUB TELL
2030 COLOUR=C( RND (3)+1)
2040 GOTO 2050+20* RND (2)
2050 X= RND (279)
2060 GOTO 2080
2070 Y= RND (159)
2080 GOSUB POINT
2090 GOSUB KEY
2100 GOSUB WIPE
2110 CALL LINE
2120 GOTO 2030
2500 REM ...SIMPLE 2-D WITH ZOOM
2510 GOSUB 2900
2520 CALL INIT
2526 PRINT : PRINT
2530 FOR W=10 TO 1 STEP -1
2540 CALL CLEAR
2550 FOR I=1 TO 56
2560 IF E(I)<20000 THEN GOTO 2580
2565 CHECK=1
2570 X=(E(I)-20000)/100
2575 GOTO 2590
2580 X=E(I)/100
2585 CHECK=0
2590 Y=E(I) MOD 100
2600 X=X*3/W+15*(W-1)
2610 Y=Y*3/W+10*(W-1)
2620 GOSUB 2800
2670 NEXT I
2680 FOR DELAY=1 TO 100
2690 DEL=DELAY*W
2700 GOSUB KEY
2710 NEXT DELAY
2720 NEXT W
2730 VTAB 23: PRINT "PRESS ANY KEY FOR MENU"
2740 GOTO 2530
2800 IF X<0 THEN X=0: IF Y<0 THEN Y=0: IF X>278 THEN X=278: IF Y>
   159 THEN Y=159
2810 GOSUB POINT
2820 IF CHECK=1 THEN GOTO 2840
2830 CALL LINE: GOTO 2850
2840 CALL POSN
2850 RETURN
2900 REM ...ENTERPRISE
2901 E(1)=20230
2902 E(2)=940
2903 E(3)=5240
2904 E(4)=5437
2905 E(5)=5730
2910 E(6)=230
2911 E(7)=1029
2912 E(8)=2228
2913 E(9)=4028
2914 E(10)=5730
2915 E(11)=4028
2916 E(12)=3826
2917 E(13)=3626
2918 E(14)=3422
2919 E(15)=2922
2920 E(16)=2726
2921 E(17)=3626
2922 E(18)=2426
2923 E(19)=2228
2924 E(20)=21940
2925 E(21)=3247
2926 E(22)=4340
```

8199 in the listing. Originally, I keyed in the two statements in Fig. 1. If I key LIST 8199,8200 then I get the combined lines as shown in the listing, but if I key LIST 8200 then I get line 8200 as shown above. In any case, the program works just fine.

I hope you enjoy the Hi-Res program as much as I have. I've found it to be a good demo program when you're showing off your Apple II to visitors. By the way, the Random Horizontals and Verticals subprogram gives quite a different effect if the paddle button is kept depressed. ■

```
2927 E(23)=4960
2928 E(24)=7060
2929 E(25)=E(4)
2930 E(26)=7060
2931 E(27)=7765
2932 E(28)=7768
2933 E(29)=8668
2934 E(30)=8665
2935 E(31)=9568
2936 E(32)=9773
2937 E(33)=7078
2938 E(34)=4878
2939 E(35)=3674
2940 E(36)=3664
2941 E(37)=3461
2942 E(38)=3477
2943 E(39)=3674
2944 E(40)=3664
2945 E(41)=4960
2946 E(42)=27765
2947 E(43)=7732
2948 E(44)=28665
2949 E(45)=8632
2950 E(46)=6432
2951 E(47)=6229
2952 E(48)=6126
2953 E(49)=6123
2954 E(50)=6220
2955 E(51)=6417
2956 E(52)=6432
2957 E(53)=6417
2958 E(54)=15017
2959 E(55)=14032
2960 E(56)=6432
2999 RETURN
3000 REM ...CLARK'S ART GALLERY
3010 CALL INIT: POKE 768,5
3020 POKE 769,0: POKE 804,0: POKE 805,3
3026 GOSUB TELL
3030 CALL CLEAR
3040 COLOUR=C( RND (3)+1)
3050 RR= RND (3):M= RND (9)+1:N= RND (25)+3
3060 GOTO 3070+20* RND (3)
3070 POKE 769,N
3080 GOTO 3200
3090 POKE 769,3
3100 M= RND (100)+2:N=M/10+1: GOTO 3200
3110 POKE 804, RND (5)
3120 POKE 805, RND (5)
3130 N= RND (4)
3200 FOR I=1 TO N
3210 IF RR#0 THEN COLOUR=C( RND (3)+1)
3220 X= RND (279):Y= RND (160)
3230 GOSUB POINT: CALL POSN
3235 MM= RND (M)+1
3240 FOR R=1 TO 64
3250 POKE 806,MM: POKE 807,R
3260 CALL SHAPE
3265 GOSUB KEY: GOSUB WIPE
3270 NEXT R
3280 NEXT I
3290 GOTO 3020
7998 PRINT "ERROR--REACHED 7998": END
7999 REM  SUBROUTINE 'POINT' TO DEFINE X, Y, AND COLOUR AS REQUIRED B
     Y MACHINE LANGUAGE PROGRAM
8000 POKE 828,COLOUR
8010 POKE 800,X MOD 255
8020 POKE 801,X/255
8030 POKE 802,Y
8040 RETURN
8100 REM ...SUBROUTINE 'FIRST' TO REMOVE INITIAL LINE
8110 COLOUR=0
8120 X= PDL (0)
8130 Y= PDL (1)
8140 GOSUB POINT
8150 CALL PLOT: CALL CLEAR
8160 RETURN
8199 REM ..:'GOSUB WIPE' TO CLEAR SCREEN BY PUSHING PADDLE BUTTON.:
     , RUN ^ IF PEEK (-16287)>127 OR PEEK (-16286)>127 THEN CALL
     CLEAR
8210 POKE -16296,0
8220 RETURN
8299 REM ...'GOSUB KEY' TO RETURN TO MENU IF ANY KEY PRESSED.
8300 IF PEEK (-16384)>127 THEN GOTO 8330
8310 POKE -16368,0
8320 RETURN
8330 POKE -16368,0
8340 GOTO 1000
8350 REM ...'GOSUB TELL' FOR WIPE/KEY INSTRUCTIONS ON SCREEN
8360 CALL -936
8370 VTAB (21)
8380 PRINT "PRESS PADDLE BUTTON TO CLEAR, OR ANY KEY FOR MENU"
8390 RETURN
9200 RETURN
10300 PRINT : PRINT "MACHINE LANGUAGE PROGRAM FROM APPLE'S   HIGH RESO
     LUTION DEMO TAPE MUST BE LOADEDPRIOR TO THIS PROGRAM."
```



Example of 2-D sketch of "Enterprise."

TAKE NOTE.

The Music Box IS HERE!

Newtech Computer Systems, a leading manufacturer of music peripherals and software for S-100 and SS-50 computers, introduces the MUSIC BOX.

The MUSIC BOX is a complete hardware/software tool that enables you to produce music and sound effects on your TRS-80.*

You can compose music, play or sing along with the computer, or just listen to your favorite tunes—up to four notes at a time, with a seven octave range. And you can make it sound like one, two, three or four different musical instruments at the same time. Or you can make all sorts of weird sound effects and noises like explosions, gun shots, "phasor" and other space war sounds—not to mention bells and whistles.

The MUSIC BOX plugs into the TRS-80 keyboard or the Expansion Interface Bus Extension. It includes a volume control, a 400 milliwatt power amp, and phono jack for easy connection to an external speaker. Software is supplied on Level II cassette. Requires a 32K RAM or larger Level II computer.

Future software developments include more pre-coded music, games with sound effects, music education, telephone tone dialing, and Morse code programs.

$249.

Complete with software and users manual. Add $3. for shipping and $1. if COD.

*TRS-80 and TRSDOS are trademarks of Tandy Corp.

✓240

NEWTECH
COMPUTER SYSTEMS, INC.

230 Clinton Street, Brooklyn, New York 11201 • (212) 625-6220

DISCOUNT SOFTWARE ✓250

CP/M®

OSBORNE AND ASSOC.
business software in CBASIC-2 — DISK WITH MANUAL / MANUAL ONLY

- # General Ledger
- # Acct. Rec./Acct. Pay.
- # Payroll w/Cost Acct.

} $59/$19 each

Buy 2 get 1 free .... $118/$57
All 3 plus CBASIC-2 .... $199/$71

CBASIC-2® .... $89/$14

DIGITAL RESEARCH
CP/M® 2.2 for TRS-80® Model II $149/$24
CP/M® 2.2 for Northstar ..... $149/$24

WORDSTAR by MicroPro .... $399/$24

WHITESMITHS
*"C" Complier .... $600/$29
*Pascal (includes "C") ...... $750/$44

SELECTOR III-C2 .... $269/$19

PEACHTREE business software in Microsoft BASIC source code.

- † General Ledger
- † Acct. Rec.
- † Acct. Pay.
- † Payroll
- † Inventory

} $399/$65 each

All 5 plus WORDSTAR .... $1995/$349

TRS-80® MODEL II

CP/M® 2.2
Latest Version $149

ELECTRIC PENCIL II
Standard Printer $249
Diablo, Qume, NEC $279

APPLE II®

VISICALC® by Personal Systems ... $122

VISA • MASTERCHARGE
ORDERS ONLY — CALL TOLL FREE
1-800-854-2003 ext. **823A**
Calif. 1-800-522-1500 ext. **823A**

For information write or call:
THE DISCOUNT SOFTWARE GROUP
1610 Argyle Ave., Bldg. 102
Los Angeles, CA 90028
(213) 461-3127

CP/M users: specify disk systems and formats. Most formats available.

Add $2.50 postage and handling per each item. California residents add 6% sales tax. Allow 2 weeks on checks. C.O.D. ok.

Prices subject to change without notice. All items subject to availability.

* — Special Bonus with order
† — Requires microsoft BASIC
# — Requires CBASIC-2
® — Mfgs. Trademark

P.S. — We want to be your software source. Give us the opportunity to beat any nationally advertised price!

BUSINESS & PROFESSIONAL SOFTWARE FOR APPLE II

☐ **HOME FINANCE PAK I: Complete package $49.95**

☐ BUDGET: The heart of a comprehensive home finance system. Allows user to define up to 30 budget items. Actual expense input can be by keyboard or by automatic reading of CHECKBOOK II files. Costs are automatically sorted and compared with budget. BUDGET produces both monthly actual/budget/variance report and a year-to-date by month summary of actual costs. Color graphics display of expenses by month. $24.95

☐ CHECKBOOK II: This extensive program keeps complete records of each check/deposit. Unique check entry system allows user to set up common check purpose and recipient categories. Upon entry you select from this pre-defined menue to minimize keying in a lot of data. Unique names can also be stored for completeness. This system produces rapid access to check files. Check register display incorporates unique up/down scrolling system for the ultimate in flexibility. 40 column printout of check register. Up to 100 checks per month storage. Files accessible by BUDGET program $14.95

☐ SAVINGS: Allows user to keep track of deposits/withdrawals for up to 10 savings accounts. Complete records shown via screen or 40 column printer. $14.95

☐ CREDIT CARD: Keep control of your cards with this program. Organizes, stores and displays purchases, payments and service charges. Screen or 40 column printer display. Up to 10 separate cards $14.95

☐ **THE UNIVERSAL COMPUTING MACHINE: $39.95**
A user programmable computing system structured around a 15 row by 30 column table. User defines row and column names and equations forming a unique computing machine. Table elements can be multiplied, divided, subtracted or added to any other element. User can define repeated functions common to a row or column greatly simplifying table setup. Hundred of unique computing machines can be defined, used, stored and recalled, with or without old data, for later use. Excellent for sales forecasts, engineering design analysis, budgets, inventory lists, income statements, production planning, project cost estimates-in short for any planning, analysis or reporting problem that can be solved with a table. Unique curser commands allow you to move to any element, change its value and immediately see the effect on other table values. Entire table can be printed by machine pages (user-defined 3-5 columns) on a 40 column printer. Transform your computer into a UNIVERSAL COMPUTING MACHINE.

☐ **COLOR CALENDAR:** HI-RES color graphics display of your personal calendar. Automatic multiple entry of repetitive events. Review at a glance important dates, appointments, anniversaries, birthdays, action dates, etc. over a 5 year period. Graphic calendar marks dates. Printer and screen display a summary report by month of your full text describing each day's action item or event. Ideal for anyone with a busy calendar $19.95

☐ **MICROACCOUNTANT:** The ideal system for the small cash business. Based on classic T-accounts and double-entry bookkeeping, this efficient program records and produces reports on account balances, general ledger journals, revenue and expenses. Screen or 40 column printer reports. Handles up to 500 journal entries per period, up to 100 accounts. Program instructions include a short primer in Financial Accounting $49.95

☐ **THE MATHEMATICIAN: Complete package $39.95**

☐ MATRIX: A general purpose utility program for computing the determinant or inverse of a matrix, as well as the solution to an N-Dimensional set of linear equations $14.95

☐ PLOTTER 2D: Two dimensional plotting of any function $9.95

☐ PLOTTER 3D: Three dimensional surface plotting of any 3-variable function, using Apple II HI-RES $14.95

☐ ROOTS: A general purpose program for numerically finding the solutions to any non-linear equation. Uses quadratic interpolation $14.95

SEND FOR FREE CATALOG — All programs run on 32K Apple II with Disc and Applesoft ROM Card, or 48K without ROM Card. Detailed instructions included with each program. California residents add 6½% sales tax. Send check or money order to:

SPECTRUM SOFTWARE ✓251
P.O. BOX 2084 – SUNNYVALE, CALIFORNIA 94087

CLASSIFIEDS

Classified advertisements are intended for use by persons desiring to buy, sell or trade used computer equipment. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows up to 5 lines of about 35 characters per line, including spaces and punctuation. The $10 box allows up to 10 lines. Minimize use of capital letters to save space. No special layouts allowed. Payment is required in advance with ad copy. We cannot bill or accept credit.

Advertising text and payment must reach us 60 days in advance of publication (i.e., copy for March issue, mailed in February, must be here by Jan. 1). The publisher reserves the right to refuse questionable or inapplicable advertisements. Mail copy with payment to: **Classifieds, Kilobaud Microcomputing, Peterborough NH 03458.** Do not include any other material with your ad as it may be delayed.

Super Elf and expansion board. All options plus super monitor and Tiny BASIC on ROM. Complete documentation, all for $190. G. Patrick, Apt. 2-9E, 531 Hahaine, Honolulu HI 96825.

Wanted: SWTP computer with dual floppies. Dr. Gordon, 1435 W. 49th Place, Hialeah FL. (305) 822-1100.

For Sale: North Star floating-point board & manual, $250. Sol & 32K memory, CRT, cassette recorder, BASIC5, ext. cassette BASIC, ASL-8, FOCAL & manuals, $1600. CUTS cassette interface rev. B & manual, $80. Two Tarbell cassette interfaces rev. D & manual, $80 each. D. G. Kissinger, School of Health, Loma Linda Univ., Loma Linda CA 92350. (714) 796-7272.

SWTP 6800 4K unit assembled and tested, no peripherals, complete with manuals—asking $400. Cdn funds. Ed Vander Ryd, P.O. Box 4166, Stn. D, Hamilton, Ontario, Canada L8V 4L5.

Need: IC, type #SW-10667 pt #14027 used in 1622 Diablo keyboard, micro switch #75SD12-4. Also KB timing diagrams. Jim Jamison, 2304 Tucker, Rte. 6, N.E., No. Ft. Myers FL 33903.

For Sale: Commodore PET 2023 friction feed printer. New/never used. Cost $849. Will sell for $700. G. H. Ashley, 25 Whitney Tavern Rd., Weston MA 02193, (617) 894-4334.

For Sale: New Apple II, disks, Applesoft ROM card, printer, D.C. Hayes modem, CRT monitor, clock calendar card, graphics input tablet, Applewriter word processor program. Sell all or part. Call Don at (714) 776-6384 or send for list of items with prices. D. Anderl, Box 20EA, Fullerton CA 92633.

For Sale: KIM-1 single board computer. Original doc. extra on expanding, etc. Must sell, $150. Wanted: 4115s from upgraded PETs 8/$20. John Lemkelde, 5980 Bull Rd., Dover PA 17315, (717) 292-4933.

Wanted: Utility billing package for TRS-80 Model II running under CP/M or TRSDOS. Send information and price to: Edwin Phillips, 928 Davis, New Madrid MO 63869.

For Sale: KIM-1 includes 1K RAM, 2K ROM, power supply, 3 manuals, plus *Programming a Micro: 6502*. Asking $125/best offer. Call Ed (617) 544-2207 in PM.

For Sale: Soroc IQ 120 terminal in original carton, $595. Don Pittman, 1816 Cleary Ave., Metairie LA 70001. (504) 837-3267.

Wanted: Ohio Scientific C-3B and/or 74 meg hard disk. And/or OSI C30EM. Bill Brown, San Juan Bocs, 201 12 Street, Durango CO 81301.

For Sale: Digital Equipment PDP8/L, $300. C. L. Jones, 2338 Raunsdale Rd., Kettering OH 45440. (513) 449-6796.

For Sale: ASR-33 Teletype with stand and full documentation, $400 or best offer. TRS-232 interface, $40. R. W. Carney, 1927 S. Dewey, Bartlesville OK 74003.

Save—new 40 track BASF 5¼" minifloppy drive model #6106 (Shugart compatible) in factory carton—$245. V. M. Faulkner, RR. 2, Box 294A, Yorktown IN 47396. (317) 289-4138.

AIM-65 for sale w/4K RAM, power supply (5 V @ 8 A, 24 V @ 2 A, plus!) Current loop & audio cables. Mint cond. $400. Call (617) 997-2811, ext. 783, ask for Paul Smola.

CALENDAR

Waltham MA

Anyone interested in learning more about the Exatron Stringy Floppy is invited to the first East Coast ESFOA Workshop, Saturday, May 31, at Micro Communications, 80 Bacon St., Waltham MA. Units available for hands-on experience; refreshments available. Workshop will run from 10 to 6. Phone (617) 899-8111 for more details or directions.

Champaign IL

An update review of existing technology and recent advances will be presented at the Ninth Annual Symposium on Incremental Motion Control Systems and Devices, June 2–5, 1980, Ramada Inn, Champaign, Illinois. For further information, contact Professor B. C. Kuo, IMCSS, PO Box 2772, Station A, Champaign IL 61820, (217) 333-4341.

Blacksburg VA

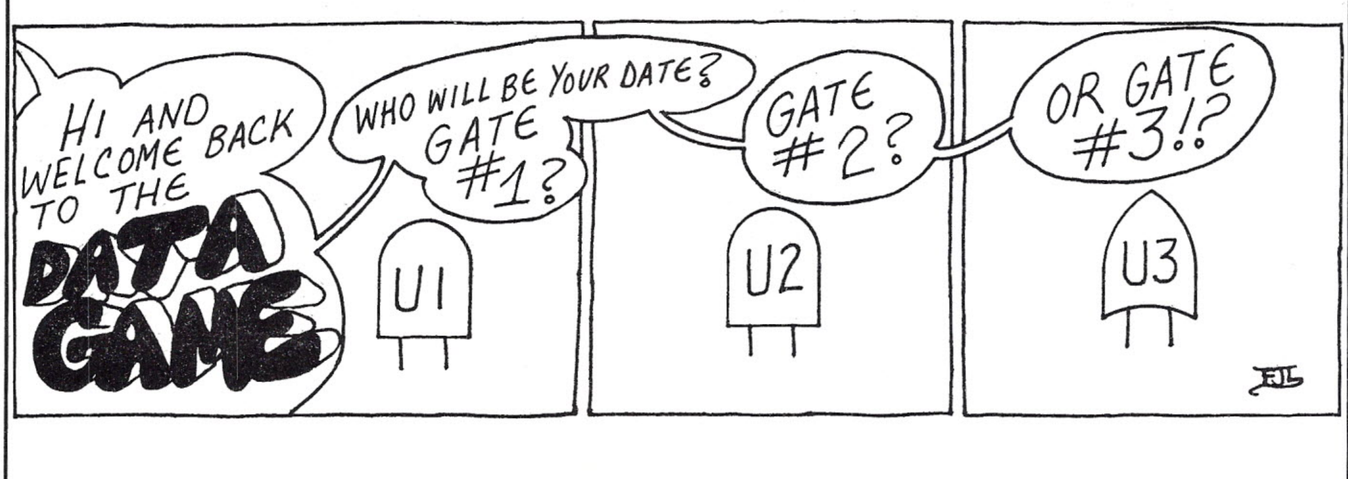
Workshop: TRS-80 Interfacing and Programming for Instrumentation and Control. June 23–27, 1980. This is a hands-on workshop with the participants working with and designing interfaces for the TRS-80. For more information, contact Dr. Linda Leffel, CEC, Virginia Tech., Blacksburg VA 24061, (703) 961-5241.

Philadelphia PA

The fifth Produx 2000 will be held May 21–23, 1980, at the Philadelphia Civic Center from 11 AM–6 PM. Contact Produx 2000, Inc., (215) 457-2300 to reserve space.

Guelph Ontario

Central Ontario Amateur Radio Flea Market and Computer Fest will be held at the Centennial Arena, College Ave. West, Guelph Ontario, Saturday, June 7, 1980, 8 AM to 4 PM. Admission only $1 (12 years and under—free). For further information, please contact Rocco Furfaro VE3HGZ, (519) 824-1157.



DEALER DIRECTORY

Hobbyists and Businessmen These dealers can supply your needs in the home and business computer market. Call the one nearest you!

Los Angeles CA
Featuring: PolyMorphic, North Star, Imsai, Cromemco, Extensys, Speechlab products and Poly-88 Users Group software exchange. All products 10-20% off list. We won't be undersold! **A-A-A-A Discount Computer How's, 1477 Barrington, Suite 17, Los Angeles CA 90025, 477-8478.**

Mountain View CA
Systems for business, industry and hobbyist. Five terminals, ten printers and five mainframes on display. Superbrain, Horizon Quad, Compucolor II, Equibox and Altos Computer System with hard disk. Much software incl. CP/M, TRS-80 and PET. **Digital Deli Computer Store, 80 W. El Camino Real, Mt. View CA 94040, 961-2670.**

North Hollywood CA
Wholesale prices to dealers & computer club members. Computers, printers, terminals, modems, boards, etc. **Patio Computer Sales Co., 5437 Laurel Canyon Bl., #208, N. Hollywood CA 91607, 762-0020.**

San Jose CA
Bay area's newest computer store. Featuring the new Texas Instruments TI 99/4 home & business computer. Software for TRS-80, Apple, PET, etc. Magazines. **Hobbi-Tronics, 1378 S. Bascom Ave., San Jose CA 95128, 998-1103.**

Pompano Beach FL
Business systems, personal systems, whatever the application, we can help. Consulting, programming, education and maintenance. Service, support and professionalism at affordable prices. **Computer Age Inc., 1308 N. Federal Hwy., Pompano Beach FL 33062, 946-4999.**

Venice FL
Discount prices & professional service: Cromemco, North Star, Vector Graphic, DEC, TI, Thinker Toys, Intertube, Soroc, Centronics, NEC, Selectric interfaces, Microdasys. Complete business & medical billing software available. MicroAge & Serendipity software discounted. **Sara-Tech Electronics, Inc., Computer Division, PO Box 692, Venice FL 33595, 485-3559.**

Aurora IL
Microcomputer systems for home or business; peripherals, software, books & magazines. Apple, Hewlett-Packard, North Star, Cromemco systems. IDS-440G printer w/Apple graphics, New HP-85 & HP calculators. **Farnsworth Computer Center, 1891 N. Farnsworth Ave., Aurora IL 60505, 851-3888.**

Chicago IL
Computer Hardware/Software Specialists for home and business. Largest selection of computer books, magazines and copyrighted software in Chicago Metro area. Experienced factory-trained service department. Feature Apple and Alpha Microsystems and accessories. **Data Domain of Schaumburg, 1612 E. Algonquin Road., Schaumburg IL 60195, 397-8700.**

Chicago IL
SS50-SS50C-6800-6809 specialists. Dealers for Gimix, Smoke Signal Broadcasting, SWTPC, TSC. Business Basic, Fast BASIC, payroll, editor-text processor. Custom programming. **AAA Chicago Computer Center, 120 Chestnut Lane, Wheeling IL 60090, 459-0450.**

Granite City IL
Come see our full line of Apple II computers & peripherals, North Star and Cromemco business systems. Full line of components, books and magazines. Ten minutes from St. Louis MO. **Computer Station, #12 Crossroads Plaza, Granite City IL 62040, 452-1860.**

Naperville IL
Computer systems design, programming and consultation by computer experts. Dealer for SSM, Integrand, Tarbell, Ithaca Intersystems, Verbatim, Diablo and others. Discount prices on many items. **Wilcox Enterprises, 25W178-39th St., Naperville IL 60540, 420-8601.**

Laurel MD
Apple & Exidy Sorcerer. Authorized dealer for both. We also sell subscriptions to "The Source" timeshare service. Plus full line of ham equipment. **The Comm Center, Laurel Plaza, Rte. 198, Laurel MD 20810, 792-0600.**

Worcester MA
Computer products for personal and business systems. Largest selection of software for TRS-80, Apple, PET. Authorized Apple sales and service. **Computer Packages Unlimited, Centerwood Terrace, Route 12, West Boylston MA 01583, 835-3428.**

Worcester MA
Complete business and personal systems on Data General, microNova and Apple. Software, books, magazines, accessories and supplies. Custom business programming available. Authorized Apple service. Downtown. **The Computer Place Inc., 11 Harvard St., Worcester MA 01609, 755-5387.**

Garden City MI
Complete systems for business, professional and personal applications. Custom programming available. Apple II, North Star, Vector Graphic and other lines of microcomputers, software, books, components. **Computer Center, 28251 Ford Rd., Garden City MI 48135, 422-2570.**

Minneapolis/St. Paul MN
Discount systems, terminals, software, peripherals & furniture for businesses, O.E.M.s, institutions & hobbyists. Send for mail-order catalog. Grand openings in two northern metro-Minneapolis locations. Z8000 systems soon! **Binary Information Systems, Box 326K, Anoka MN 55303, 757-5878.**

St. Louis MO
Experimenters' Paradise. Electronic and mechanical components. Computer People, Audio People, Hams, Robot Builders, Experimenters. Open six days a week. **Gateway Electronics Corp., 8123-25 Page Blvd., St. Louis MO 63130, 427-6116.**

New York NY
S-100 New York City systems provided for business and office applications. We also seek experienced S-100 programmers on a part-time basis. **Olcott International & Co., PO Box 3014, Weehawken NJ 07087, (212) 863-4200.**

Rome-Utica NY
Ohio Scientific authorized dealer. Computers for personal, business, education and industry. Software for word processing, education, business, personal and games. **Comp-u-Ware, 9057 Main St., Westernville NY 13486, 827-4487.**

Akron OH
We've got it all. Business systems. Personal systems. Software packages. Custom programming. Terminals. Printers. Service and books. Easy freeway access. 10 AM to 6 PM Monday–Saturday. **The Basic Computer Shop, Fairlawn Plaza, 2671 West Market St., Akron OH 44313, 867-0808.**

Portland OR
Ohio Scientific specialists for business and personal computers. Local service. Terminals, printers, custom programming. Full OSI product line on display! 10AM to 6PM M–F. **Fial Computer, 11266 SE 21st Ave., Milwaukie OR 97222, 654-9574.**

Kingston PA
We support Level II and Model II. Books, magazines, programs, parts, accessories, peripherals, free literature, free seminars, cassettes, floppies, filters, transformers, caps, chips, CRTs. **Artco Electronics, 302 Wyoming Ave., Kingston PA 18704, 287-1014.**

Philadelphia/So. Jersey
North Star, fast delivery possible; word-processing systems, Heath terminals, Intertube II, immediate delivery. Free video terminal comparison, Intertec's SuperBrain, all Centronics printers, Omnitec data modems. **L & S Distributors, 44 So. Locust, Marlton NJ 08053, 983-7444.**

Sara Mexico
Learn how to utilize and program the Z-80 microprocessor to maintain your company's records in top shape. This technique has just been introduced to Mexico. Courses, applications, maintenance, service. **Digitales, S.A. de C.V. Sara 4612, Mexico 14, D.F., 5-17-41-59.**

Dealers: Listings are **$15** per month in prepaid quarterly payments, or one yearly payment of **$150**, also prepaid. Ads include 25 words describing your products and services plus your company name, address and phone. (No area codes or merchandise prices, please.) Call Marcia at 603-924-7138 or write *Kilobaud MICROCOMPUTING*, Ad Department, Peterborough NH 03458.

Reader Service—see page 258

kb microcomputing book nook

NEW ADDITIONS

● **PROGRAMMING THE Z-80** — BK1122 — by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

● **6502 ASSEMBLY LANGUAGE PROGRAMMING** — BK1176 — by Lance A. Leventhal. This book provides comprehensive coverage of the 6502 microprocessor assembly language. Leventhal covers over 80 programming examples from simple memory load loops to complete design projects. Features include 6502 assembler conventions, input/output devices and interfacing methods, and programming the 6502 interrupt system. $12.50.*

● **TRS-80 DISK AND OTHER MYSTERIES** — BK1181 — by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50.*

● **BASIC BASIC (2ND EDITION)** — BK1026 — by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. $9.45.*

● **LEARNING LEVEL II** — BK1175 — by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*

● **THE INCREDIBLE SECRET MONEY MACHINE** — BK1178 — by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. $5.95.*

● **FREELANCE SOFTWARE PUBLISHING** — BK1179 — by B. J. Korites. "This book is about money and how to make it by writing and selling computer programs," (author's foreword). If you have the skills to write a saleable program, you now need to acquire the skills to sell that program. This compact book comprehensively covers the entire publishing process and many aspects of software salesmanship. $14.95.*

● **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1045 — by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $14.95.*

● **6502 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1055 — by Robert Findley. This book introduces the BASIC language programmer into the realm of machine-language programming. The description of the 6502 structure and instruction set, various routines, subroutines and programs are the ingredients in this cookbook. "Recipes" are included to help you put together exactly the programs to suit your taste. $10.95.*

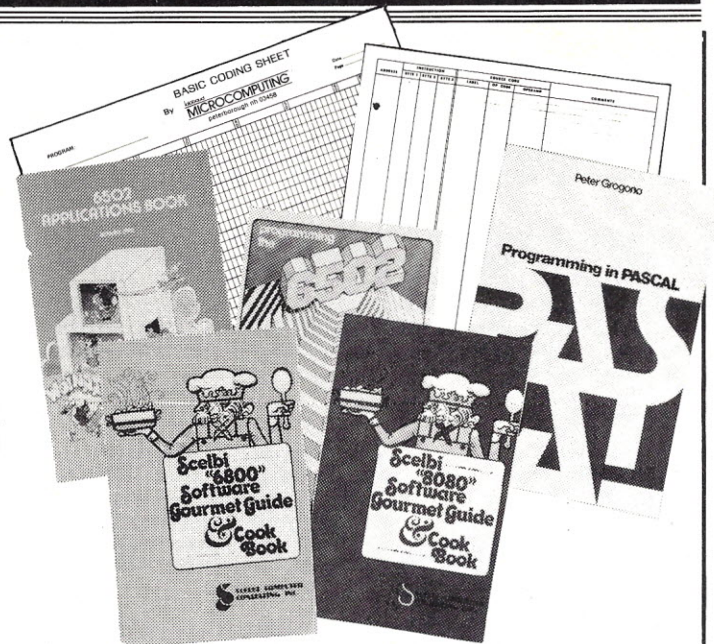
● **INTRODUCTION TO TRS-80 GRAPHICS** — BK1180 — by Don Inman. Dissatisfied with your Level I or Level II manual's coverage of graphics capabilities? This well-structured book (suitable for classroom use) is ideal for those who want to use all the graphics capabilities built into the TRS-80. A tutorial method is used with many demonstrations. It is based on the Level I, but all material is suitable for Level II use. $8.95.*

● **Z-80 ASSEMBLY LANGUAGE PROGRAMMING** — BK1177 — by Lance A. Leventhal. This book thoroughly covers the Z80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $12.50.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook



● **PROGRAMMING IN PASCAL** – BK1140 – by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such, it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language; it is therefore suitable for an introductory course. $11.50.*

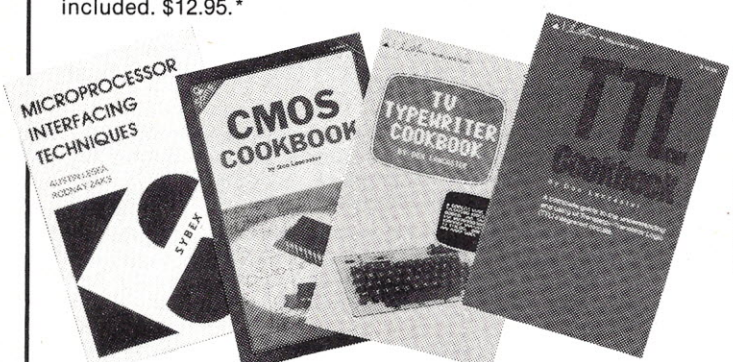
● **8080 SOFTWARE GOURMET GUIDE AND COOKBOOK** – BK1102 – If yu have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other routines for the 8080 user. $10.95.*

● **6800 SOFTWARE GOURMET GUIDE AND COOKBOOK** – BK1075 – Like its culinary cousin, *The 8080 Gourmet Guide*, this new book by Scelbi Computing and Robert Findley describes sorting, searching and other routines – this time for the 6800 user. $10.95.*

● ***MICROCOMPUTING* CODING SHEETS** *Microcomputing's* dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is **Assembly/Machine Language** (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for **BASIC** (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

● **PROGRAMMING THE 6502 (Second Edition)** – BK1005 – Rodnay Zaks has designed a self-contained text to learn programming, using the 6502. It can be used by a person who has never programmed before, and should be of value to anyone using the 6502. The many exercises will allow you to test yourself and practice the concepts presented. $12.95.*

● **6502 APPLICATIONS BOOK** – BK1006 – Rodnay Zaks presents practical-application techniques for the 6502 microprocessor, assuming an elementary knowledge of microprocessor programming. You will build and design your own domestic-use systems and peripherals. Self-test exercises included. $12.95.*

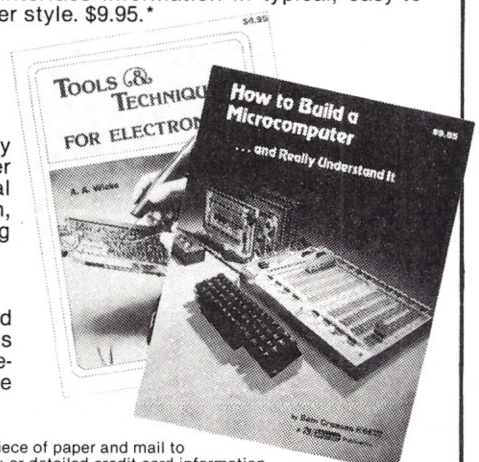


● **MICROPROCESSOR INTERFACING TECHNIQUES** – BK1037 – by Austin Lesea & Rodnay Zaks will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S100. $15.95.*

● **TTL COOKBOOK** – BK1063 – by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter and a digital tachometer. $9.50.*

● **CMOS COOKBOOK** – BK1011 – by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $10.50.*

● **TVT COOKBOOK** – BK1064 – by Don Lancaster. Describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.*



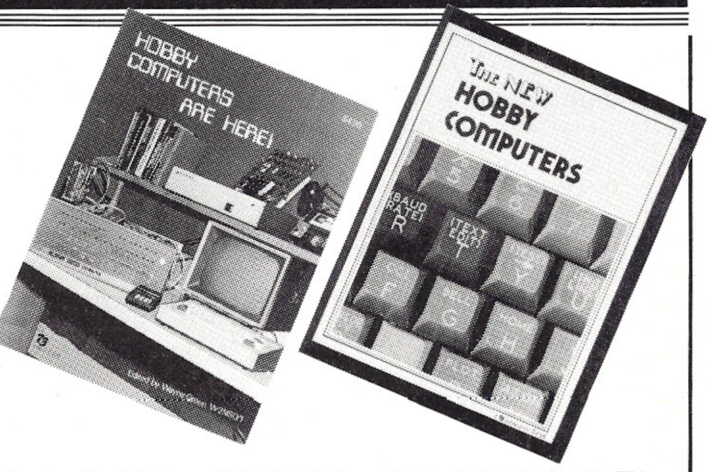
● **HOW TO BUILD A MICROCOMPUTER – AND REALLY UNDERSTAND IT** – BK7325 – by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95.*

● **TOOLS & TECHNIQUES FOR ELECTRONICS** – BK7348 – is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects as well as specialized metal working tools and the chemical aids which are used in repair shops. $4.95.*

* Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

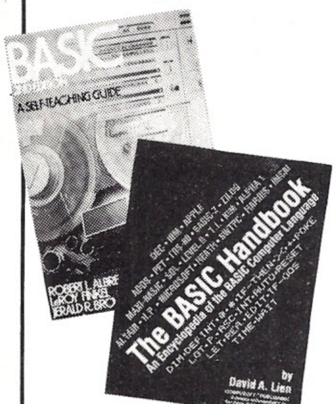
FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook



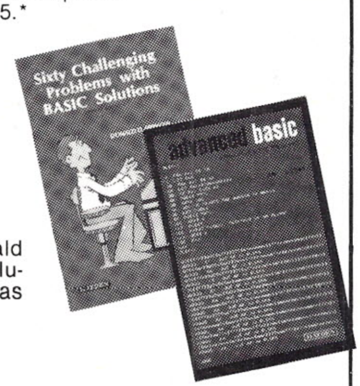
●**HOBBY COMPUTERS ARE HERE!**—BK7322—If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, and the basics of programming. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

●**THE NEW HOBBY COMPUTERS**—BK7340—This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards . . . and much, much more! Don't miss this tremendous value! Only $4.95.*



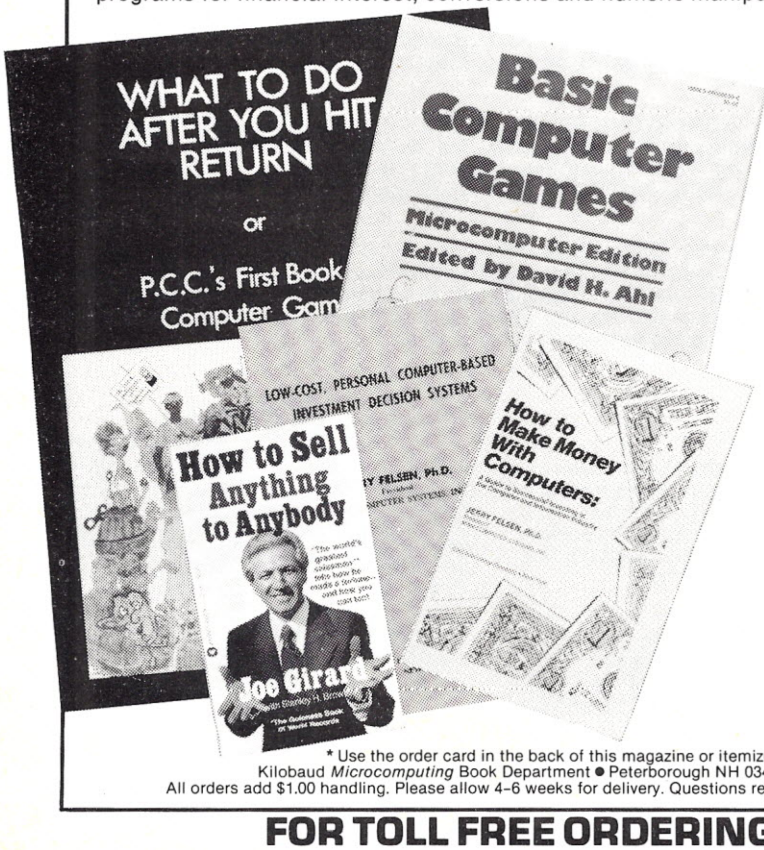
●**THE BASIC HANDBOOK**—BK1174—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

●**BASIC NEW 2ND EDITION**—BK1081—by Bob Albrecht. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $6.95.*



●**ADVANCED BASIC**—BK1000—Applications and problems by James Coan is for those who want to extend their expertise with BASIC. Offers advanced techniques and applications. $9.65.*

●**SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS (2nd Edition)**—BK1073—by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Includes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*



●**LOW-COST, PERSONAL COMPUTER-BASED INVESTMENT DECISION SYSTEMS**—BK1101—Use this guidebook by Man-Computer Systems, Inc.'s president, Jerry Felsen, to develop inexpensive personal computer systems that can help you make better investment decisions. $15.00.*

●**HOW TO SELL ANYTHING TO ANYBODY**—BK7306—According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune—and how you can, too. $2.25.*

●**HOW TO MAKE MONEY WITH COMPUTERS**—BK1003—In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

●**WHAT TO DO AFTER YOU HIT RETURN**—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

●**BASIC COMPUTER GAMES**—BK1074—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

* Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

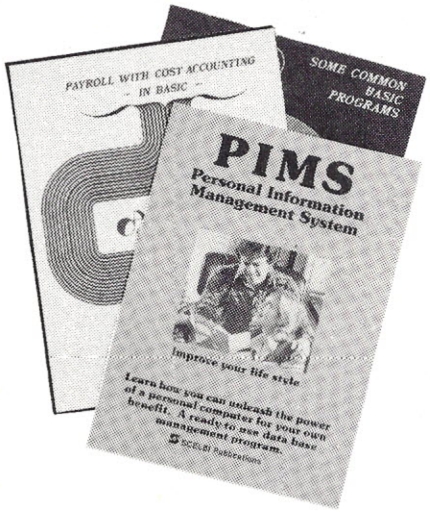
kb microcomputing book nook

●AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0—BK1130—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. $7.95.*

●VOL. I—BK1030—Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. $8.50.*

●VOL. II—BK1031 (loose leaf)—$25.00*; BK1040 (with binder)—$30.00*—Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips.

●VOL. III—BK1132 (loose leaf)—$15.00*; BK1133 (with binder)—$20.00.* Contains descriptions of all support devices that can be used with any microprocessor.



TOLL FREE ORDERING CALL 1-800-258-5473

●PAYROLL WITH COST ACCOUNTING—IN BASIC—BK1001— by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941. $20.00.*

●SOME COMMON BASIC PROGRAMS—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $12.50 paperback.*

●PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM—BK1009—Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $9.95.*

* Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

only a few left...

1977 KILOBAUD: the COMPLETE COLLECTION for $20

That's right, we're liquidating the last issues of *Kilobaud* we have been saving . . . these are COLLECTOR'S COPIES . . . with several of the individual copies worth more than we're asking for the complete year.

This includes the very hard-to-get #1 issue which has been going for as much as $25 at auctions . . . and a library shelf box . . . the whole works, complete. When these rare copies are gone, that will be the end of it. This is your last chance to get a complete collection of Kilobaud . . . starting from the first

During this first year *Kilobaud* had 224 feature articles, including 85 with complete program listings. Just as a source of programs this is a bargain.

The $20 price includes the library shelf box and shipping charges within the USA. You can call our toll free number and order this special: (800) 258-5473. Have your credit card handy.

Please specify #KB1977 when ordering.

P.O. Box 4430S, Santa Clara, CA 95054
Will calls: 2322 Walsh Ave.
(408) 988-1640

Quest ELECTRONICS

✓44

NEW PRODUCTS!

Super Color S-100 Video Kit $99.95

Expandable to 256 x 192 high resolution color graphics. 6847 with all display modes computer controlled. Memory mapped. 1K RAM expandable to 6K. S-100 bus 1802, 8080, 8085, Z80 etc.

Gremlin Color Video Kit $59.95

32 x 16 alpha/numerics and graphics; up to 8 colors with 6847 chip; 1K RAM at E000. Plugs into Super Elf 44 pin bus. Not expandable to high resolution Graphics.

Elf II Adapter Kit $24.50

Plugs into Elf II providing Super Elf 44 and 50 pin bus plus S-100 bus expansion (With Super Expansion). High and low address displays, state and mode LED's optional **$18.00.**

1802 16K Dynamic RAM Kit $149.00

1802/S-100 expandable to 32K, Hidden refresh w/clocks up to 4 MHz w/no wait states Addl. 16K RAM **$79.00.**

Quest Super Basic

Quest, the leader in inexpensive 1802 systems announces another first. Quest is the first company worldwide to ship a **full size Basic** for 1802 systems. A complete function **Super Basic** by **Ron Cenker** including floating point capability with scientific notation (number range ±.17E38), 32 bit integer ±2 billion; Multi dim arrays; String arrays; String manipulation; Cassette I/O, Save and load, Basic, Data and machine language programs; and over 75 Statements, Functions and Operators.

Easily adaptable on most 1802 systems. Requires 12K RAM minimum for Basic and user programs. Cassette version in stock now. ROM versions coming soon with exchange privilege allowing some credit for cassette version.

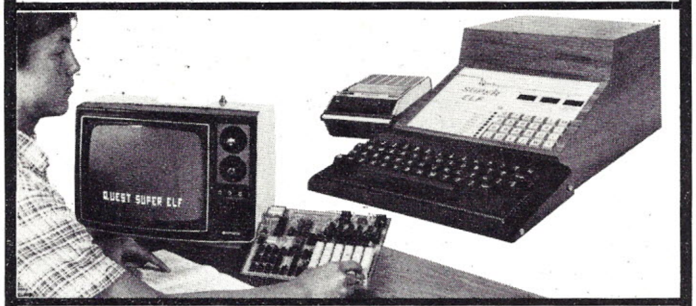
Super Basic on Cassette **$40.00**

Tom Pittman's 1802 Tiny Basic Source listing now available. Find out how Tom Pittman wrote Tiny Basic and how to get the most out of it. Never offered before. **$19.00**

S-100 4-Slot Expansion **$ 9.95**

Super Monitor VI.I Source Listing **$15.00**

Coming Soon: Assembler, Editor, Disassembler, DA/AD, Super Sound/Music, EPROM programmer, Stringy Floppy Disc System.



RCA Cosmac Super Elf Computer $106.95

Compare features before you decide to buy any other computer. There is no other computer on the market today that has all the desirable benefits of the **Super Elf** for so little money. The Super Elf is a small single board computer that does many **big** things. It is an excellent computer for training and for learning programming with its machine language and yet it is easily **expanded with additional memory, Full Basic, ASCII Keyboards, video character generation, etc.**

Before you buy another small computer, see if it includes the following features: ROM monitor; State and Mode displays; Single step; Optional address displays; Power Supply; Audio Amplifier and Speaker; Fully socketed for all IC's; Real cost of in warranty repairs; Full documentation.

The **Super Elf** includes a **ROM** monitor for program loading, editing and execution with **SINGLE STEP for program debugging** which is not included in others at the same price. With **SINGLE STEP** you can see the microprocessor chip operating with the **unique Quest** address and data bus displays **before, during** and **after** executing instructions. Also, CPU mode and instruction cycle are decoded and displayed on 8 LED indicators.

An RCA 1861 **video graphics** chip allows you to connect to your own TV with an inexpensive video modulator to do graphics and games. There is a **speaker system** included for writing your own music or using many music programs already written. The speaker amplifier may also be used to drive relays for control purposes.

A 24 key **HEX keyboard** includes 16 HEX keys plus **load, reset, run, wait, input, memory protect, monitor select** and **single step**. Large, on board displays provide output and optional **high and low address**. There is a 44 pin standard connector slot for PC cards and a 50 pin connector slot for the Quest Super Expansion Board. Power supply and sockets for **all** IC's are included in the price plus a detailed 127 pg. instruction manual which now includes over 40 pgs. of software info. including a series of lessons to help get you started and a music program and graphics target game. Many schools and universities are using the Super Elf as a course of study. OEM's use it for training and R&D.

Remember, other computers only offer Super Elf features at additional cost or not at all. **Compare before you buy. Super Elf Kit $106.95, High address option $8.95, Low address option $9.95. Custom Cabinet** with drilled and labelled plexiglass front panel **$24.95.** Expansion Cabinet with room for 4 S-100 boards **$41.00. NiCad Battery Memory Saver Kit $6.95.** All kits and options also completely assembled and tested.

Questdata, a 12 page monthly software publication for 1802 computer users is available by subscription for $12.00 per year. Issues 1-12 bound $16.50.

Tiny Basic Cassette **$10.00,** on ROM **$38.00,** original Elf kit board **$14.95. 1802 software;** Moews Video Graphics **$3.50.** Games and Music **$3.00,** Chip 8 Interpreter **$5.50.**

Super Expansion Board with Cassette Interface $89.95

This is truly an astounding value! This board has been designed to allow **you** to decide how you want it optioned. **The Super Expansion Board comes with 4K of low power RAM** fully addressable anywhere in 64K with built-in memory protect and a **cassette interface.** Provisions have been made for all other options on the same board and it fits neatly into the hardwood cabinet alongside the **Super Elf.** The board includes slots for up to 6K of **EPROM** (2708, 2758, 2716 or TI 2716) and is **fully socketed.** EPROM can be used for the monitor and Tiny Basic or other purposes.

A IK Super ROM Monitor $19.95 is available as an on board option in 2708 EPROM which has been preprogrammed with a program loader/editor and error checking multi file cassette read/write software, (relocatable cassette file) another exclusive from Quest. It includes register save and readout, block move capability and video graphics driver with blinking cursor. Break points can be used with the register save feature to isolate program bugs quickly, then follow with single step. The Super Monitor is written with subroutines allowing users to take advantage of monitor functions simply by calling them up. Improvements and revisions are easily done with the monitor. If you have the **Super Expansion Board** and **Super Monitor** the monitor is up and running at the push of a button.

Other on board options include **Parallel Input and Output Ports with full handshake.** They allow easy connection of an ASCII keyboard to the input port. **RS 232 and 20 ma Current Loop** for teletype or other device are on board and if you need more memory there are two **S-100** slots for static RAM or video boards. Also a 1K Super Monitor version 2 with video driver for full capability display with Tiny Basic and a video interface board. **Parallel I/O Ports $9.85, RS 232 $4.50, TTY 20 ma I/F $1.95, S-100 $4.50. A 50 pin connector set** with ribbon cable is available at $15.25 for easy connection between the **Super Elf** and the **Super Expansion Board.**

Power Supply Kit for the complete system (see Multi-volt Power Supply below).

Same day shipment. First line parts only. Factory tested. Guaranteed money back. Quality IC's and other components at factory prices.

INTEGRATED CIRCUITS

7400TTL	
7400N	.17
7402N	.18
7404N	.19
7409N	.23
7410N	.18
7414N	.70
7420N	.18
7422N	.39
7430N	.20
7442N	.50
7445N	.69
7447N	.60
7448N	.69
7450N	.18
7474N	.35
7475N	.49
7485N	.88
7489N	1.85
7490N	.43
7492N	.43
7493N	.43
7495N	.69
74100N	.90
74107N	.35
74121N	.34
74123N	.59
74125N	.45
74145N	.69
74150N	.95
74151N	.69
74154N	1.00
74157N	.69
74161N	.87
74162N	.87
74163N	.87
74174N	.96
74175N	.90
74190N	1.15
74192N	.87
74193N	.85
74221N	2.00
74298N	1.65
74365N	.66
74366N	.66
74367N	.66
74LS00 TTL	
74LS00N	.35
74LS02N	.35
74LS04N	.35
74LS05N	.35
74LS08N	.35
74LS10N	.35
74LS13N	.55
74LS14N	1.10
74LS20N	.35
74LS22N	.35
74LS28N	.41
74LS30N	.35
74LS33N	.75
74LS38N	.55
74LS74N	1.25
74LS75N	1.00
74LS90N	.85
74LS93N	.70
74LS95N	1.10
74LS107N	.45
74LS112N	.45
74LS113N	.35
74LS132N	.89
74LS136N	.45
74LS151N	.85
74LS155N	.85
74LS157N	.85
74LS162N	1.15
74LS163N	1.15
74LS174N	2.00
74LS190N	1.06
74LS221N	1.95
74LS258N	.67
74LS367N	1.35
LINEAR	
CA3045	.90
CA3046	1.10
CA3081	1.80
CA3082	1.90
CA3089	2.95
LM301AN/AH	.35
LM305H	.87
LM307N	.35
LM308N	.89
LM309K	1.50
LM311H/N	.90
LM317T/K	3.75
LM318	1.35
LM320K-5	1.50
LM323K-5	5.95
LM320K-12	1.50
LM320K-15	1.50
LM320T-5	1.60
LM320T-8	1.60
LM320T-12	1.50
LM320T-15	1.60
LM324N	1.15
LM339N	1.55
LM340K-5	1.35
LM340K-8	1.35
LM340K-12	1.35
LM340K-15	1.35
LM340K-24	1.35
LM340T-5	1.25
LM340T-8	1.25
LM340T-12	1.25
LM340T-15	1.25
LM340T-18	1.25
LM340T-24	1.25
LM343H	4.50
LM350	7.50
LM370	1.15
LM377	3.00
LM379	5.00
LM380N	1.00
LM381	1.60
LM382	1.60
LM703H	.40
LM709H	.28
LM723H/N	.50
LM733N	.67
LM741CH	.35
LM741N	.32
LM747H/N	.75
LM748N	.35
LM1303N	1.75
LM1304	1.10
LM1305	1.27
LM1307	2.00
LM1310	2.75
LM1458	.47
LM1800	1.75
LM1812	7.50
LM1889	3.00
LM2111	1.75
LM2902	1.50
LM3900N	.60
LM3905	1.75
LM3909N	.89
MC1458V	.50
NE540H	3.75
NE550N	1.00
NE555V	.39
NE556A	.85
NE565A	1.00
NE566V	1.50
NE567V	1.00
NE570B	5.00
78L05	.60
78L08	.60
78M05	.85
75108	1.75
75491CN	.50
75492CN	.55
75494CN	.89
A to D CONVERTER	
8038B	4.50
8700CJ	13.95
8701CN	22.00
8750CJ	13.95
LD130	9.95
9400CJV/F	7.40
ICL7103	9.50
ICL7107	14.25
CMOS	
CD34001 Fair.	.50
CD4000	.16
CD4001	.28
CD4002	.28
CD4006	1.10
CD4007	.28
CD4008	.28
CD4009	.45
CD4010	.45
CD4011	.28
CD4012	.28
CD4013	.39
CD4014	1.00
CD4015	1.00
CD4016	.45
CD4017	1.05
CD4018	.94
CD4019	.45
CD4020	1.02
CD4021	1.25
CD4022	1.10
CD4023	.28
CD4024	.75
CD4025	.28
CD4026	2.00
CD4027	.66
CD4028	.85
CD4029	1.02
CD4030	.45
CD4035	1.02
CD4040	1.02
CD4042	.85
CD4043	.85
CD4044	.85
CD4046	1.67
CD4049	.45
CD4050	.49
CD4051	1.13
CD4060	1.42
CD4066	.71
CD4068	.40
CD4069	.40
CD4070	.50
CD4071	.28
CD4072	.28
CD4073	.28
CD4075	.28
CD4076	1.45
CD4078	.40
CD4081	.28
CD4082	.28
CD4116	.47
CD4490	5.50
CD4507	1.00
CD4508	4.25
CD4510	1.02
CD4511	.94
CD4515	2.52
CD4516	1.10
CD4518	1.02
CD4520	1.02
CD4527	1.51
CD4528	.79
CD4553	3.50
CD4566	2.25
CD4583	4.50
CD4585	1.10
CD40192	3.00
74C00	.28
74C04	.40
74C10	.28
74C14	2.10
74C20	.28
74C30	.28
74C48	1.95
74C74	.75
74C76	1.40
74C90	1.15
74C93	1.40
74C154	3.00
74C160	1.44
74C175	1.35
74C192	1.65
74C221	2.00
74C905	6.00
74C906	.75
74C914	1.95
74C922	5.50
74C923	5.50
74C925	6.95
74C926	6.95
74C927	6.95
INTERFACE	
8095	.65
8096	.65
8097	.65
8098	.65
8T09	1.25
8T10	4.50
8T13	3.00
8T20	5.50
8T23	3.10
8T24	3.50
8T25	3.20
8T26	1.69
8T28	2.75
8T97	1.69
8T98	1.69
MOS/MEMORY	
RAM	
2101-1	3.95
2102-1	.95
2102AL-4	1.25
2102AN-2L	1.60
21L02-1	1.18
2104A-4	4.95
2107B-6	3.75
2111-1	3.75
2112-2	3.95
2114L 300ns	7.40
4116 200ns	9.50
8/4116 200ns	64.00
2513B	6.30
MM5262	.40
MM5280	3.00
MM5320	9.95
MM5330	5.94
PD411D-3	4.00
PD411D-4	5.00
P5101L	8.95
4200A	9.95
82S25	2.90
91L02A	1.50
HD0165-5	6.95
MM57100	4.50
GIAY38500-1	9.95
MCM66751A	9.95
9368	3.50
410D	10.00
416	16.00
CLOCKS	
MM5311	5.50
MM5312	3.90
MM5314	3.90
MM5369	2.10
MM5841	14.45
MM5865	7.95
CT7001	5.80
CT7010	8.95
CT7015	8.95
MM5375AA/N	3.90
MM5375AG/N	4.90
7205	16.50
7207	7.50
7208	15.95
7209	4.95
DS0026CN	3.75
DS0056CN	3.75
MM53104	2.50
MICROPROCESSOR	
6502	12.50
6504	16.50
6522	13.60
6800	17.50
6802	18.75
6820	9.95
6850	12.95
8080A	8.95
8085	27.00
8086	75.00
Z80	14.75
Z80A	19.75
8212	2.90
8214	8.00
8216	2.90
8224	2.90
8228	5.35
8251	8.50
8253	15.00
8255	9.25
8257	19.50
8259	19.50
1802CP plas.	13.95
1802DP plas.	17.95
1861P	11.50
CDP1802CD	19.95
CDP1802D	25.00
CDP1861	12.95
UART/FIFO	
AY5-1013	5.50
AY5-1014	7.50
3341	6.95
PROM	
1702A	3.95
2513B upper case	6.95
2708	7.75
2716T1	24.50
2716 Intel	34.95
8/2716 Intel	240.00
2732	85.00
2758	22.50
8741A	85.00
8748	75.00
8748-8	70.00
8755A	65.00
N82S23	2.95
N82S123	3.50
N82S126	3.75
N82S129	3.75
N82S131	3.75
N82S136	8.75
N82S137	8.75
DM8577	2.90
8223	2.90
8/2716 Intel	240.00

CONNECTORS

30 pin edge	2.50
44 pin edge	2.75
100 pin edge	4.50
100 pin edge WW	5.25

IC SOCKETS
Solder Tin Low Profile

PIN	1 UP	PIN	1UP
8	.15	22	.30
14	.14	24	.35
16	.16	28	.42
18	.27	36	.58
20	.29	40	.57

2 level 14 pin ww .20

WIRE WRAP LEVEL 3

PIN		PIN	
14	.32	24	.86
16	.33	28	1.00
18	.57	40	1.23

CRYSTALS

1 MHz	4.50
2 MHz	4.50
4 MHz	4.25
5 MHz	4.25
10 MHz	4.25
18 MHz	3.90
20 MHz	3.90
32 MHz	3.90
32768 Hz	4.00
1.8432 MHz	4.50
3.5795 MHz	1.20
2.0100 MHz	1.95
2.097152 MHz	4.50
2.4576 MHz	4.50
3.2768 MHz	4.50
5.0688 MHz	4.50
5.185 MHz	4.50
5.7143 MHz	4.50
6.5536 MHz	4.50
14.31818 MHz	4.25
18.432 MHz	4.50
22.1184 MHz	4.50

KEYBOARD ENCODERS

AY5-2376	$12.50
AY5-3600	17.95
AY5-9100	10.50
AY5-9200	16.50
74C922	5.50
74C923	5.50
HD0165-5	6.95

D Connectors RS232

DB25P	2.95
DB25S	3.95
Cover	1.50
RS232 Complete Set	6.50
DE9S	1.95
DA15P	2.10
DA15S	3.10

Hickok 3½ Digit LED multimeter **59.95**
Stopwatch Kit **26.95**
Auto Clock Kit **17.95**

Not a Cheap Clock Kit $14.95
Includes everything except case. 2-PC boards. 6-.50" LED Displays. 5314 clock chip, transformer, all components and full instructions. Orange displays also avail. **Same kit w/.80" displays.** Red only. **$21.95** Case **$11.75**

8K/16K Eprom Kit (less PROMS) $89.00
Motherboard $39.00
Extender Board $8.99

RESISTORS ¼ watt 5%
10 per type .03 — 1000 per type .012
25 per type .025 — 350 piece pack
100 per type .015 — 5 per type 6.75
½ watt 5% per type .05

KEYBOARDS

56 key ASCII keyboard kit	$67.50
Fully assembled	77.50
53 key ASCII keyboard kit	60.00
Fully assembled	70.00
Enclosure	14.95
Metal Enclosure	29.95

LEDS

Red T018	.15
Green, Yellow T018	.20
Jumbo Red	.20
Green, Orange, Yellow Jumbo	.25
Cliplite LED Mounting Clips	8/$1.25

(specify red, amber, green, yellow, clear)

CONTINENTAL SPECIALTIES in stock
Complete line of breadboard test equip.
MAX-100 8 digit Freq. Ctr. **$128.95**

OK WIRE WRAP TOOLS in stock
Portable Multimeter **$18.00**

SPECIAL PRODUCTS

MM5865 Stopwatch Timer with 10 pg. spec.	9.00
PC board	7.50
Switches Mom. Pushbutton	.27
3 pos. slide	.25
Encoder HD0165-5	6.95
Paratronics 100A Logic Analyzer Kit	**$224.00**
Model 10 Trigger Expander Kit	$229.00
Model 150 Bus Grabber Kit	$369.00
Sinclair 3½ Digit Multimeter	**$59.95**
Clock Calendar Kit	**$23.95**
2.5 MHz Frequency Counter Kit	**$37.50**
30 MHz Frequency Counter Kit	**$47.75**

TRANSFORMERS

6V 300 ma	3.25
12 Volt 300 ma transformer	1.25
12.6V CT 600 ma	3.75
12V 250 ma wall plug	2.95
12V CT 250 ma wall plug	3.50
24V CT 400 ma	3.95
10V 1.2 amp wall plug	4.85
12V 6 amp	12.95
12V 500 ma wall plug	4.75
12V 1 amp wall plug	6.50
12V 3 amp	8.50
10/15 VAC 8/16 VA wall plug	9.75

DISPLAY LEDS

MAN1	CA	.270	2.90
MAN3	CC	.125	.39
MAN72/74	CA/CA	.300	1.00
DL704	CC	.300	1.25
DL707/DL707R	CA	.300	1.00
DL727/728	CA/CC	.500	1.90
DL747/750	CA/CC	.600	1.95
DL750	CC	.600	1.95
FND359	CC	.357	.70
FND500/507	CC/CA	.500	1.35
FND503/510	CC/CA	.500	.90
FND800/807	CC/CA	.800	2.20
3 digit Bubble			.60
4 digit Bubble			.80
DG8 Fluorescent			1.75
DG10 Fluorescent			1.75
5 digit 14 pin display			1.00
10 digit display			1.25
7520 Clairex photocells			.39
TIL311 Hex			9.50
MAN3640	CC	.30	1.10
MAN4610	CA	.40	1.20
MAN4640	CC	.40	1.20
MAN4710	CA	.40	.95
MAN4740	CC	.40	1.20
MAN6640	CC	.56	2.95
MAN6710	CA	.60	1.35
MAN6740	CC	.60	1.35
MA1002A			**8.95**
MA1002E			**8.95**
MA1012A			**8.95**
102P3 transformer			**2.25**

ROCKWELL AIM 65 Computer

6502 based single board with full ASCII keyboard and 20 column thermal printer. 20 char. alphanumeric display, ROM monitor, fully expandable. $375.00. 4K version $450.00. 4K Assembler $85.00, 8K Basic Interpreter $100.00.

Special small power supply for AIM65 assem. in frame **$49.00.** Complete AIM65 in thin briefcase with power supply **$485.00.** Molded plastic enclosure to fit AIM65 plus power supply **$47.50.** Special Package Price: 4K AIM, 8K Basic, power supply, cabinet **$599.00**

AIM65/KIM/VIM/Super Elf 44 pin expansion board; 3 female and 1 male bus. Board plus 3 connectors **$22.95.**

AIM65/KIM/VIM I/O Expansion Kit; 4 parallel and 2 serial ports plus 2 internal timers **$39.00.** PROM programmer for 2716 **$150.00.**

Multi-volt Computer Power Supply

8v 5 amp, ±18v .5 amp, 5v 1.5 amp, −5v .5 amp, 12v .5 amp, −12 option. ±5v, ±12v are regulated. Kit $29.95. Kit with punched frame $37.45, $4.00 shipping. Kit of hardware $14.00. Woodgrain case $10.00, $1.50 shipping.

PROM Eraser Will erase 25 PROMs in 15 minutes. Ultraviolet, assembled **$37.50**
Safety switch/Timer version **$69.50**

60 Hz Crystal Time Base Kit $4.40

Converts digital clocks from AC line frequency to crystal time base. Outstanding accuracy.

NiCad Battery Fixer/Charger Kit

Opens shorted cells that won't hold a charge and then charges them up, all in one kit w/full parts and instructions. **$7.25**

LRC 7000+Printer $389.00

40/64 column dot matrix impact, std. paper. Interface all personal computers.

Televideo Terminal $845.00

102 key, upper, lowercase, 10 Baud rates 24 x 80 char. microprocessor cont. edit. cap.

Intertube II Terminal $874.00

Super Brain

Floppy Disk Terminal $2895.00

79 IC Update Master Manual $29.95

Complete IC data selector, 2500 pg. master reference guide. Over 50,000 cross references. Free update service through 1979. Domestic postage $3.50. No foreign orders.

S-100 Computer Boards

8K Static RAM Kit	$135.00
16K Static RAM Kit	265.00
24K Static RAM Kit	423.00
32K Static RAM Kit	475.00
16K Dynamic RAM Kit	199.00
32K Dynamic RAM Kit	310.00
64K Dynamic RAM Kit	470.00
Video Interface Kit	**$129.00**

Video Modulator Kit $8.95

Convert TV set into a high quality monitor w/o affecting usage. Comp. kit w/full instruc.

Digital Temp. Meter Kit $34.00

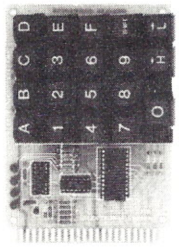
Indoor and outdoor. Switches back and forth. Beautiful. 50" LED readouts. Nothing like it available. Needs no additional parts for complete, full operation. Will measure −100° to +200°F, tenths of a degree, air or liquid. Beautiful woodgrain case w/bezel **$11.75**

**TERMS: $5.00 min. order U.S. Funds. Calif residents add 6% tax.
BankAmericard and Master Charge accepted.
Shipping charges will be added on charge cards.**

FREE: Send for your copy of our NEW 1980 QUEST CATALOG. Include 28¢ stamp.

✔ Reader Service—see page 258

HEX ENCODED KEYBOARD

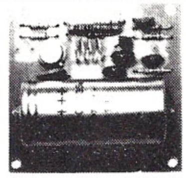


Four onboard LEDs indicate the HEX code generated for each key depression. The board requires a single +5 volt supply. Board only $15.00 Part No. HEX-3, with parts $49.95 Part No. HEX- 3A. 44 pin edge connector $4.00 Part No. 44P.

ASCII TO CORRESPONDENCE CODE CONVERTER

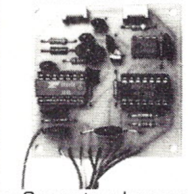
This bidirectional board is a direct replacement for the board inside the Trendata 1000 terminal. The on board connector provides RS-232 serial in and out. Sold only as an assembled and tested unit for $249.95. Part No. TA 1000C

T.V. INTERFACE



• Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple • Power required is 12 volts AC C.T., or +5 volts DC • Board only $7.60 part No. 107, with parts $13.50 Part No. 107A

TAPE INTERFACE

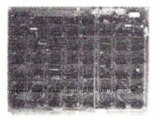


• Converts a low cost tape recorder to a digital recorder • Works up to 1200 baud • Digital in and out are TTL-serial • Output of board connects to mic. in of recorder • Earphone of recorder connects to input on board • No coils • Requires +5 volts, low power drain • Board only $7.60 Part No. 111, with parts $29.95 Part No. 111A

ASCII KEYBOARD

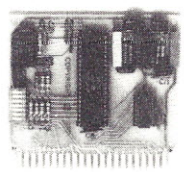
53 Keys popular ASR-33 format • Rugged G-10 P.C. Board • Tri-mode MOS encoding • Two-Key Rollover • MOS/DTL/TTL Compatible • Upper Case lockout • Data and Strobe inversion option • Three User Definable Keys • Low contact bounce • Selectable Parity • Custom Keycaps • George Risk Model 753. Requires +5, -12 volts. $59.95 Kit.

T.V. TYPEWRITER



• Stand alone TVT • 32 char/line, 16 lines, modifications for 64 char/line included • Parallel ASCII (TTL) input • Video output • 1K on board memory • Output for computer controlled curser • Auto scroll • Non-destructive curser • Curser inputs: up, down, left, right, home, EOL, EOS • Scroll up, down • Requires +5 volts at 1.5 amps, and -12 volts at 30 mA • All 7400, TTL chips • Char. gen. 2513 • Upper case only • Board only $39.00 Part No. 106, with parts $145.00 Part No. 106A

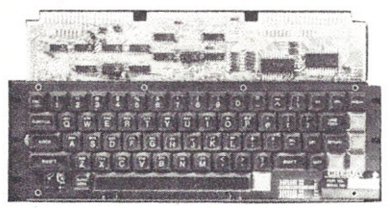
UART & BAUD RATE GENERATOR



• Converts serial to parallel and parallel to serial • Low cost on board baud rate generator • Baud rates: 110, 150, 300, 600, 1200, and 2400 • Low power drain +5 volts and -12 volts required • TTL compatible • All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity. • All connections go to a 44 pin gold plated edge connector • Board only $12.00 Part No. 101, with parts $35.00 Part No. 101A, 44 pin edge connector $4.00 Part No. 44P

ASCII KEYBOARD

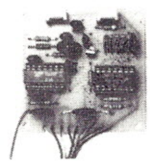
TTL & DTL compatible • Full 67 key array • Full 128 character ASCII output • Positive logic with outputs resting low • Data Strobe • Five user-definable spare keys • Standard 22 pin dual card edge connector • Requires +5VDC, 325 mA. Assembled & Tested. Cherry Pro Part No. P70-05AB. $119.95.



SOROC IQ 120

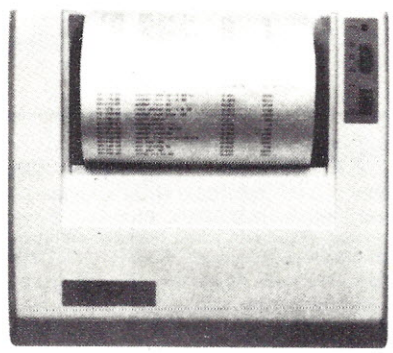
Upper/lower case display • Numeric keypad & cursor keys • Protected fields, ½ intensity display • RS 232 interface & aux. port. IQ120—$799.95 • IQ140 Detachable keyboard—$1199.95

MODEM



• Type 103 • Full or half duplex • Works up to 300 baud • Originate or Answer • Serial TTL input and output • connect 8 Ω speaker and crystal mic. directly to board • Requires +5 volts • Board only $7.60 Part No. 109, with parts $29.95 Part No. 109A.

COMPRINT PRINTER

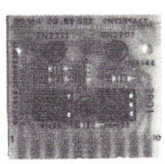


Printing Characteristics: 225 characters/second (170 lines/minute) throughput • 9 horizontal x 12 vertical matrix • 96 ASCII character set with upper and true lower case • 80 characters/line • 5.8 lines/inch
Buffer Memory: standard 256 bytes; • optional; 2,048 bytes (buffer memory option designated as Model 912-2K), add $149.95.
Paper Requirements: electrosensitive type (aluminum coated) • 8-1/2 inch width • 3.7 inch max. (300 ft.) roll diameter.
Model 912-S Interfacing: serial interface RS232 and 20 mA current loop • BAUD rates 110, 150, 300, 600, 1200, 2400 and 4800 are strap selectable.
Model 912-P Interfacing: parallel interface, IEEE-488 and 8 bit parallel (strobe/ acknowledge). Model 912-S, Part No. CPIA, 32118, $579.95. Model 912-P, Part No. CPIA, 32117, $559.95.

44 BUS MOTHER BOARD

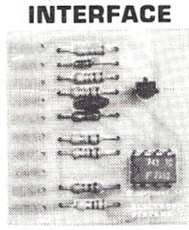
Has provisions for ten 44 pin (.156) connectors, spaced 3/4 of an inch apart. Pin 20 is connected to X, and 22 is connected to Z for power and ground All the other pins are connected in parallel. This board also has provisions for bypass capacitors. Board cost $15.00 Part No. 102. Connectors $3.00 each Part No. 44WP.

RS-232/20mA INTERFACE



This board has two passive, opto-isolated circuits. One converts RS-232 to 20mA, the other converts 20mA to RS-232. All connections go to a 10 pin edge connector. Requires +12 and -12 volts. Board only $9.95, part no. 7901, with parts $14.95 Part No. 7901A.

RS-32/TTL INTERFACE



• Converts TTL to RS-232, and converts RS-232 to TTL • Two separate circuits • Requires -12 and +12 volts • All connections go to a 10 pin edge connector, kit $9.95 Part No. 232A 10 Pin edge connector $3.00 part No. 10P.

COMPUCOLOR II

With reg. keyboard
MOD3 8K $1595.95
MOD4 16K $1695.95
MOD 5 32K $1995.95
Now includes $250 more, worth of software and accessories with 101 key option add $134.95 with 117 key option add $179.95

DC POWER SUPPLY

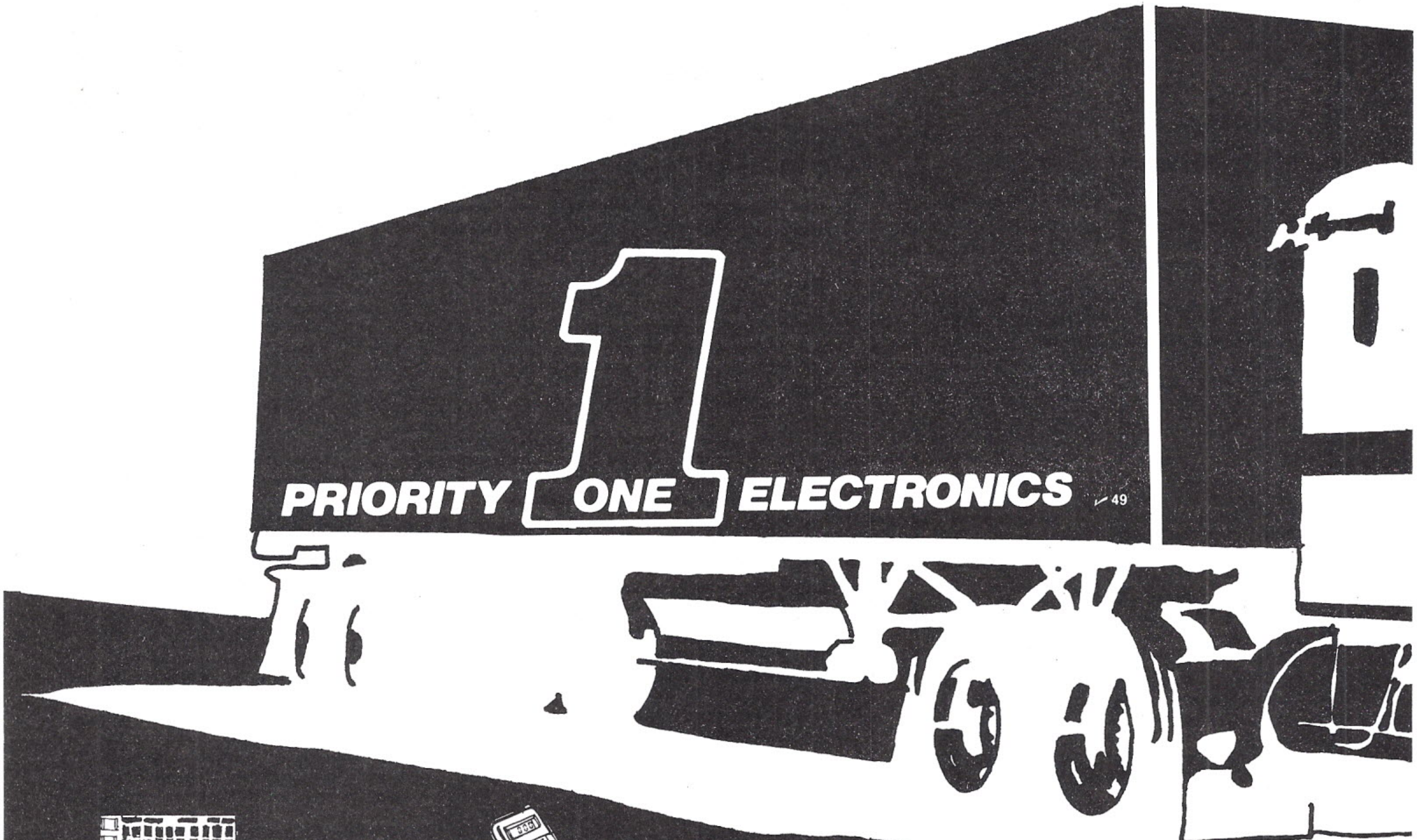
• Board supplies a regulated +5 volts at 3 amps., +12, -12, and -5 volts at 1 amp. • Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps. • Board only $12.50 Part No. 6085, with parts excluding transformers $42.50 Part No. 6085A

To Order: Mention part no. description, and price. In USA shipping paid by us for orders accompanied by check or money order. We accept C.O.D. orders in the U.S. only, or a VISA or Master Charge no., expiration date, signature, phone no., shipping charges will be added. CA residents add 6.5% for tax. Outside USA add 10% for air mail postage and handling. Payment must be in U.S. dollars. Dealer inquiries invited. 24 hour order line (408) 448-0800 ✓47

Send for FREE Catalog . . . a big self-addressed envelope with 41¢ postage gets it fastest!

ELECTRONIC SYSTEMS Dept. KB P. O. Box 21638, San Jose, CA USA 95151

✓Reader Service—see page 258



CCS-M-XVI-M

The true 16K Static Ram module for S-100 bus systems. ASSEMBLED & TESTED--100% BURN IN · FEATURES: • Fully static • Uses popular 2114 static RAMS • +5 volt operation only • Bank Select available by bank port and bank byte • Phantom Line capability • Addressable in 4K blocks • 4K blocks can be addressed anywhere with in 64K in 4k increments. • Meets IEEE proposed S-100 signal standards • LED indicators for board selection and bank selection • FR-4 EPOXY PC boards • Solder masked on both sides • Silk screen of part number and part designator.

OUR PRICE
CCS-2016BA 450ns 2MHZ ·
Reg. Price $349.95 . . . . . $279.95
CCS-2016BB 300ns 4MHZ ·
Reg. Price · $389.95 . . . . . $309.00
CCS-2016BY Bare Board · only . . . $29.95

The VISTA V-80 Disk Drive System

• 23% more storage capacity than TRS-80 • 120 day warranty • 40 track patch at NO CHARGE from VISTA

Single drive system $395.00
Two drive system $770.00
Four drive system $1450.00
Two drive cable $29.95
Four drive cable $39.95

NEW MS-230 DUAL TRACE MINISCOPE 30 MHz BANDWIDTH

1 NLS MS-230 30 MHZ Scope . . . $598.15
2 NLS 41-141 10 to 1 Combo Probe $54.00
1 NLS 41-180 Deluxe Leather Case $45.00
LIST PRICE $697.15
MS 230 COMBO PRICE $547.15
SAVE $150.00

AUDIO RESPONSE™ DIGITAL DISPLAY

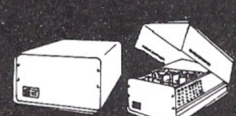
The ROAD RUNNER offers Audio Response...A new dimension in multimeters. Rapid checking of fuses, capacitors, diodes, resistors, transistors, LED's darlingtons...without looking at the meter. The <1V and<2V ranges cause sufficient voltage across the probe tips to forward bias diodes and transistors causing a continuous tone. Similarly, capacitors can be charged until the tone shuts off...thus the approximate value of the capacitor can be measured. The <1 ohm,<10 ohm, and<100 ohm ranges are used in threshold testing of resistors and making continuity tests of PCB's, wire wrap and solder connections, light bulbs, and transformer windings. In addition, the Audio Response function permits the monitoring of incoming digital pulses. The ROAD RUNNER ADMM ™ FEATURES • Six Functions • 29 Ranges • 0.5% Accuracy on DCV • 5 Range Audio Response Function • Color coded easy-to-read front panel and pushbuttons • Full Accessory Line Expands Capability • 0.5" LCD Display • Rugged Case for "Field Use" • RFI Shielded.

~~ONLY $139.00~~ $119.00

WESTON-6100
Digital Multimeter . . . . @ $139.00
WESTON-6100C Leather Case @ $ 15.95
WES-9740-0279627 VHF RF
Voltage Probe . . . . @ $ 63.00
WES-5768 AC Clamp-on Probe @ $ 57.75
WES-1961 RF Voltage Probe . . @ $ 29.00
WES-9395 50 KV Probe . . . . . @ $ 38.75
WES-6100-AC 110V
AC Adapter . . . . @ $ 15.00

NEW FROM SAMS BOOKS

192 PAGES TRS-80 . . . . . $8.95



CALIFORNIA Computer Systems

• S-100 compatible • Industrial/commercial quality construction • Flip-top cover • Excellent cooling capability • 12 slot capability (uses model 2501A) • Input 105, 115, or 125 VAC • Output + 8 VDC20A, + − 16 VDC 4A • Active termination of all bus lines • Fan and circuit breaker included • Rugged construction

CCS-2200A Assembled & Tested
35 lbs $399.95
CCS-2200AK Kit 35 lbs. $349.95

MEMORY MEMORY

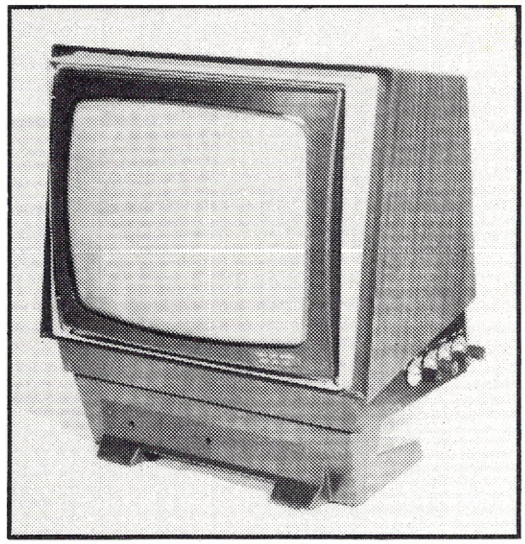
2102LIPC
Low Power 450ns in lots of 20 . . . . $1.10
2102AL-2
Low Power 250ns in lots of 20 . . . . $1.25
2114-3L
1Kx4 300ns Low Power . . . . . . 8/$50.00
5257-3L
4Kx1 300ns Low Power . . . . . . 8/$55.00
2708
8K 450ns EPROM . . . 8/$60.00-$8.50 ea.
2716
16K 5 Volt only EPROM
8/$248.00 . . . . $32.00 ea.

RS232 and "D" SUB-MINIATURE CONNECTORS *
P Plug, Male Type S Socket, Female Type C Cover, Hood

PART NO.	DESCRIPTION	PRICE 1-9	10-24	25-99
CND-DE9P	9 Pin Male	1.60	1.40	1.30
CND-DE9S	9 Pin Female	2.25	2.00	1.90
CND-DE9C	9 Pin Cover	1.50	1.35	1.20
CND-DA15P	15 Pin Male	2.35	2.15	2.00
CND-DA15S	15 Pin Female	3.25	3.10	2.90
CND-DA15C	15 Pin Cover	1.60	1.45	1.30
CND-DB25P	25 Pin Male	2.80	2.60	2.40
CND-DB25S	25 Pin Female	3.60	3.40	3.20
CND-DB51212-1	1 pc.Grey Hood	1.50	1.30	1.10
CND-DB110963-3	2 pc.Grey Hood	1.75	1.50	1.35
CND-DB1226-1A	2 pc.Black Hood	1.90	1.65	1.45

SELECTRONICS

Does it again . . . high quality Sylvania monitors at the lowest prices ever. These monitors have been thoroughly checked and guaranteed.

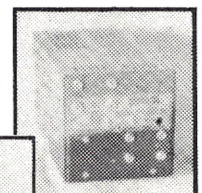


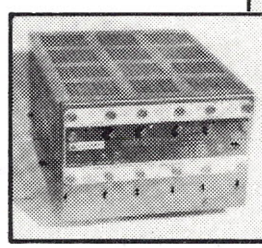
MONITORS

12" black & white monitor. Wide band, will display 80 x 24 char. 10K or 75Ω input impedance, composite video input. Transformer power supply.
Shpt. Wt. 30#
Price: $45.00 ea.
Used with all computers such as: TRS-80, Apple, and many others.
Complete manual $3.00

REGULATED DC POWER SUPPLIES
MFGS. LAMBDA & NORTH

VOLTS	– AMPS	WT.	PRICE
5	74	62#	$40.00
5	31	40	35.00
5	16	18	30.00
5	10	18	25.00
5	4	7	20.00
5	13	20	30.00
5	20	30	35.00





CABLES

5' RG/59U cable with PL259 connector on one end.
Price: $1.00 ea. 6/$5.00

24" RG/59U cable with PL 259 connector on one end, BNC on other end.
Price: $1.00 ea. 6/$5.00

POWER TRANSFORMERS, CRW/UTC APD988

5¼" x 4⅜" x 5", 17 lbs. open frame, diagram of hookup furnished, 110 or 220 volt input, output: 15 V @ 2 amps, 15 V @ 2 amps, 16 V Ct @ 5 amps, 22 V @ 3 amps, 22 V Ct @ 10 amps.

$10.00 each or 3 for $25.00

- Test Equipment
- Power Supply Components
- Power Supplies
- Communication Equipment
- Pulse Equipment

ALL ITEMS ARE REMOVED FROM EQUIPMENT

ALL ITEMS FOB OUR WAREHOUSE

SELECTRONICS

✓54

TELEPHONES:
Area Code 215
HOward 8-4645
HOward 8-7891

WAREHOUSE
1206-18 S. Napa Street
1201-49 S. Patton Street
1207-25 S. Napa Street

1206 S. Napa Street ● Philadelphia PA 19146

Penna. resident please add 6% sales tax. ALL PRICES ARE F.O.B. our warehouse, Phila. PA. All merchandise accurate as to description to the best of our knowledge. Your purchase money refunded if not satisfied. Min. order $10.00.

✓ Reader Service—see page 258

ADVANCED COMPUTER PRODUCTS

7400

SN7400N	.19	SN74123N	.59
SN7401N	.22	SN74125N	.39
SN7402N	.22	SN74126N	.44
SN7403N	.22	SN74128N	.59
SN7404N	.22	SN74132N	.69
SN7405N	.23	SN74136N	.95
SN7406N	.23	SN74139N	.95
SN7407N	.23	SN74141N	.69
SN7408N	.26	SN74142N	2.95
SN7409N	.23	SN74143N	2.95
SN7410N	.22	SN74144N	2.95
SN7411N	.29	SN74145N	.62
SN7412N	.29	SN74147N	1.95
SN7413N	.39	SN74148N	1.20
SN7414N	.59	SN74150N	.99
SN7416N	.29	SN74151N	.67
SN7417N	.29	SN74152N	.67
SN7420N	.22	SN74153N	.67
SN7421N	.35	SN74154N	1.19
SN7422N	.29	SN74155N	.82
SN7423N	.29	SN74156N	.89
SN7425N	.29	SN74157N	.69
SN7426N	.29	SN74158N	1.65
SN7427N	.29	SN74160N	.95
SN7429N	.45	SN74161N	.95
SN7430N	.23	SN74162N	.89
SN7432N	.29	SN74163N	.87
SN7437N	.29	SN74164N	.97
SN7438N	.29	SN74165N	.97
SN7439N	.29	SN74166N	1.20
SN7440N	.24	SN74167N	1.95
SN7441N	.79	SN74170N	1.69
SN7442N	.57	SN74172N	5.95
SN7443N	.79	SN74173N	.79
SN7444N	.79	SN74174N	.89
SN7445N	.79	SN74175N	.89
SN7446N	.79	SN74176N	.85
SN7447N	.59	SN74177N	.85
SN7448N	.79	SN74179N	1.80
SN7450N	.23	SN74180N	.75
SN7451N	.23	SN74181N	1.75
SN7453N	.23	SN74182N	.75
SN7454N	.23	SN74184N	1.95
SN7459N	.29	SN74185N	1.95
SN7460N	.23	SN74186N	9.95
SN7470N	.39	SN74188N	3.90
SN7472N	.34	SN74190N	1.15
SN7473N	.38	SN74191N	1.15
SN7474N	.36	SN74192N	.85
SN7475N	.38	SN74193N	.85
SN7476N	.36	SN74194N	.85
SN7479N	4.60	SN74195N	.85
SN7480N	.59	SN74196N	.85
SN7481N	1.10	SN74197N	.85
SN7482N	1.10	SN74198N	1.39
SN7483N	.55	SN74199N	1.39
SN7485N	.88	SN74221N	1.39
SN7486N	.39	SN74251N	.95
SN7489N	1.75	SN74273N	1.05
SN7490N	.39	SN74279N	.89
SN7491N	.65	SN74283N	2.15
SN7492N	.52	SN74284N	3.90
SN7493N	.49	SN74285N	3.90
SN7494N	.72	SN74290N	1.25
SN7495N	.65	SN74298N	.95
SN7496N	.72	SN74365N	.68
SN7497N	3.10	SN74366N	.68
SN74100N	.99	SN74367N	.79
SN74107N	.32	SN74368N	.79
SN74109N	.53	SN74390N	1.90
SN74116N	1.95	SN74393N	1.90
SN74121N	.29	SN74490N	1.90
SN74122N	.39		

CMOS

CD4000	.29	CD4093	.99
CD4001	.29	CD4094	2.95
CD4002	.29	CD4098	2.49
CD4006	1.39	CD4099	2.25
CD4007	.29	MC14408	12.95
CD4008	1.39	MC14409	12.95
CD4009	.59	MC14410	12.95
CD4010	.59	MC14412	12.95
CD4011	.29	MC14415	8.95
CD4012	.29	MC14419	4.95
CD4013	.49	CD4501	.39
CD4014	1.39	CD4502	1.65
CD4015	1.39	CD4503	.69
CD4016	.59	CD4505	8.95
CD4017	1.19	CD4506	.75
CD4018	1.19	CD4507	.95
CD4019	.49	CD4508	3.95
CD4020	1.19	CD4510	1.39
CD4021	1.49	CD4511	1.39
CD4022	1.29	CD4512	1.39
CD4023	.38	CD4515	3.95
CD4024	.79	CD4516	1.69
CD4025	.38	CD4518	1.39
CD4027	.79	CD4520	1.39
CD4028	.99	CD4555	4.95
CD4029	1.29	CD4556	.99
CD4030	.69	CD4566	2.25
CD4031	3.25	74C00	.39
CD4032	2.15	74C02	.39
CD4034	3.25	74C04	.39
CD4035	1.19	74C08	.49
CD4037	1.95	74C10	.49
CD4040	1.29	74C14	1.65
CD4041	1.25	74C20	.39
CD4042	.99	74C30	.39
CD4043	.99	74C32	.99
CD4044	.99	74C42	1.85
CD4046	2.25	74C48	2.39
CD4047	1.25	74C73	.99
CD4048	.69	74C74	.99
CD4049	.69	74C85	2.49
CD4050	.69	74C89	4.95
CD4051	1.10	74C90	1.85
CD4052	1.10	74C93	1.85
CD4053	1.10	74C95	1.85
CD4055	3.95	74C107	1.19
CD4056	2.95	74C151	2.49
CD4059	9.95	74C154	3.50
CD4060	1.39	74C157	2.10
CD4066	.89	74C160	2.39
CD4069	.35	U74CC161	2.30
CD4070	.69	74C163	2.39
CD4071	.35	74C164	2.39
CD4072	.35	74C173	2.59
CD4073	.35	74C174	2.75
CD4075	.35	74C175	2.75
CD4076	1.29	74C192	2.39
CD4077	.35	74C193	2.39
CD4078	.35	74C195	2.39
CD4081	.35	74C922	7.95
CD4082	.35	74C923	6.95
CD4085	1.95	MM80C95	1.50
CD4089	2.95	MM80C97	1.25

74LS00

74LS00N	.35	74LS164N	1.19
74LS01N	.28	74LS165N	.89
74LS02N	.28	74LS166N	2.48
74LS03N	.28	74LS168N	1.89
74LS04N	.39	74LS169N	1.89
74LS05N	.28	74LS170N	1.99
74LS08N	.39	74LS173N	.89
74LS09N	.39	74LS174N	.99
74LS10N	.28	74LS175N	.99
74LS11N	.39	74LS181N	2.20
74LS12N	.39	74LS190N	1.15
74LS13N	.47	74LS191N	1.15
74LS14N	1.25	74LS192N	.98
74LS15N	.39	74LS193N	.98
74LS20N	.26	74LS194N	1.15
74LS21N	.38	74LS195N	.95
74LS22N	.38	74LS196N	.89
74LS26N	.39	74LS197N	.89
74LS27N	.39	74LS221N	1.49
74LS28N	.39	74LS240N	2.99
74LS30N	.26	74LS241N	2.49
74LS32N	.39	74LS242N	2.29
74LS37N	.79	74LS243N	2.29
74LS38N	.39	74LS244N	2.95
74LS40N	.26	74LS245N	8.95
74LS42N	.79	74LS247N	1.10
74LS47N	.79	74LS248N	1.10
74LS48N	.79	74LS249N	1.69
74LS51N	.26	74LS251N	1.79
74LS54N	.35	74LS253N	.98
74LS55N	.35	74LS257N	.98
74LS73N	.45	74LS258N	.98
74LS74N	.59	74LS259N	2.95
74LS75N	.68	74LS260N	.69
74LS76N	.45	74LS261N	2.49
74LS78N	.65	74LS266N	.59
74LS83AN	.99	74LS273N	1.75
74LS85N	1.19	74LS275N	4.40
74LS86N	.45	74LS279N	.59
74LS90N	.75	74LS283N	1.10
74LS92N	.75	74LS290N	1.29
74LS93N	.75	74LS293N	1.95
74LS95N	.88	74LS295N	1.10
74LS96N	.98	74LS298N	1.29
74LS107N	.45	74LS324N	1.75
74LS109N	.45	74LS347N	1.95
74LS112N	.49	74LS348N	1.95
74LS113N	.49	74LS352N	1.65
74LS114N	.55	74LS353N	1.65
74LS122N	.55	74LS363N	1.49
74LS123N	1.19	74LS365N	.99
74LS124N	1.35	74LS366N	.99
74LS125N	.89	74LS367N	.99
74LS126N	.89	74LS368N	.99
74LS132N	.79	74LS373N	2.75
74LS136N	.59	74LS374N	2.75
74LS138N	.89	74LS375N	.69
74LS139N	.89	74LS377N	1.95
74LS145N	1.25	74LS385N	1.95
74LS148N	1.49	74LS386N	.65
74LS151N	.79	74LS390N	1.95
74LS153N	.79	74LS393N	1.95
74LS154N	2.49	74LS395N	1.70
74LS155N	1.19	74LS399N	2.95
74LS156N	.99	74LS424N	2.95
74LS157N	.99	74LS668N	1.75
74LS158N	.75	74LS670N	2.29
74LS160N	.98	81LS95N	1.99
74LS161N	1.15	81LS96N	1.99
74LS162N	.98	81LS97N	1.99
74LS163N	.98	81LS98N	1.99

LINEAR

78H05	5.95	LM1414N	1.90
78M06	1.49	LM1458CN/N	.49
78M.G.	1.49	MC1488N	1.49
LM105H	.99	MC1489N	1.49
LM108AH	2.95	LM1496N	.89
LM300H	.79	LM1556N	1.50
LM301CN/H	.35	LM1800N	.79
LM304H	.98	LM1820N	.95
LM305H	.89	LM1850N	.95
LM306H	3.25	LM1889N	3.95
LM307CN/H	.29	LM2111N	1.75
LM308CN/H	.98	LM2900N	.99
LM309K	1.49	LM2901N	2.50
LM310CN	1.25	LM2917N	2.95
LM311D/CN/H	.98	CA3013T	2.29
LM312H	1.75	CA3018T	1.99
LM317T	2.75	CA3021T	3.49
LM318CN/H	1.49	CA3023T	2.99
LM319N/H	1.25	CA3035T	2.75
LM320K-XX*	1.49	CA3039T	1.49
LM320T-XX*	1.25	CA3046T	1.29
LM320H-XX*	1.25	LM3053N	1.49
LM323K	4.95	CA3059N	3.25
LM324N	1.25	CA3060N	3.25
LM339N	.95	CA3062N	4.95
LM340K-XX*	1.49	LM3065N	1.49
LM340T-XX*	1.25	CA3080N	1.29
LM340H-XX*	1.25	CA3081N	1.69
LM344H	1.95	CA3082N	1.69
LM348N	1.85	CA3083N	1.99
LM358CN	.98	CA3086N	1.29
LM360N	1.49	CA3089N	2.75
LM372N	1.95	CA3096N	2.49
LM376N	3.75	CA3097N	1.99
LM377N	3.75	CA3130T	2.49
LM380CN/N	1.25	CA3140T	2.49
LM381N	1.79	CA3146N	2.49
LM383T	1.95	CA3160T	1.49
LM386N	1.49	CA3190N	1.95
LM387N	1.49	CA3401N	.69
LM390N	1.95	MC3423N	1.49
NE531V/T	3.75	MC3460N	3.95
NE555V	.39	SG3524N	3.95
NE556N	.98	CA3600N	3.50
NE561T	19.95	LM3900N	.59
NE562B	7.95	LM3905N	1.49
NE565N/H	1.25	LM3909N	.98
NE566H/V	1.75	RC4131N	2.95
NE567V/H	1.50	RC4136N	1.10
NE592N	2.75	RC4151N	4.50
LM702H	2.99	LM4194	4.95
LM709N/H	.29	RC4195	4.40
LM710N/H	.98	ULN2001	1.25
LM711N/H	.39	ULN2003	1.50
LM715N	1.95	SN75450N	.59
LM723N/H	.75	SN75451N	.49
LM733N/H	.98	SN75452N	.49
LM739N	1.15	SN75453N	.49
LM741CN/H	.33	SN75454N	.49
LM741CN-14	.19	SN75491N	.89
LM747N/H	.79	SN75492N	.89
LM748N/H	.39	SN75493N	.89
LM760CN	2.95	SN75494N	.89
LM1310N	1.90		

FLOPPY DISK DRIVES

MPI B51-5¼", 40 tracks 279.00
Shugart SA400-5¼" 35 tracks 295.00
Shugart 800/801R 8" 475.00
Siemens Shugart Compatible Model FDD-120-8D 429.00
PERSCI Model 277 Dual 1195.00
WANGO/SIEMENS 5¼" Drive 290.00
MPI B52 5¼" Dual 395.00
WANGO/SIEMENS 282 Dual 5¼" 395.00
WANGO/SIEMENS 82 .... 290.00

HICKOK LX303 $69.95

.5%, 3½ digit 19 Range DVM, ½" LCD displays runs 200 hrs on 1 battery. 10 Meg Ohm Input. 1 yr. guarantee, made in U.S.A., test leads included.

Available Accessories

RC-3 115V AC Adapter $7.50
CC-3 Deluxe Padded Vinyl Carrying Case $7.50
VP-10 X10 DCV Probe Adapter/ Protector 10 Kv $14.95
VP-40 40 Kv DC Probe $35.00
CS-1 10 Amp Current Shunt $14.95

TEXAS INSTRUMENTS INCORPORATED
Calculator Discount Prices

TI 30 SP 17.95
TI 50 .... 39.95
TI 55 .... 35.95
TI 57 .... 54.95
TI 58C 97.95
TI 59 .... 219.95
TI PC 100C 146.95
TI 1035 .... 15.95
TI 1750 .... 19.95
TI 5025 .... 69.95
TI 5040 .... 79.95
TI Data Card 26.95
TI Data Chron 37.95
TI Business Analyst II 42.95
TI Programmer 48.95
TI MBA 57.95
TI Business Card .... NEW 34.95
TI Speak & Spell 64.95
TI Dataman 19.95
TI Spelling B 27.95
TI Lil' Professor 12.95
TI Vowel Power 12.95
TI Super Stumpers 4, 5, 6 .... 12.95
TI Super Stumpers 7, 8 .... 12.95

MONITORS

Sanyo 9" $169.95
Sanyo 15" 279.00
Leedex 12" 139.95
Motorola 12" High Resolution, 22 MHz, OEM Model #M3000-340 .... 219.00
Zenith 13" Color Monitor 499.00
MGA 13" Color TV 349.00
VAMP 19" Color Monitor 575.00
VAMP 15" Color Monitor 449.00

TRS-80/APPLE
MEMORY EXPANSION KITS
4116's
16K (200/250 ns.)
8 pcs. for **$62.95**
w/instructions & jumpers.

Call For Volume Pricing

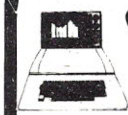
SALE $749.00 HAZELTINE TERMINALS

Model 1400 ... $749.00 | Model 1500 ... $1085.00
Model 1410 ... $825.00 | Model 1510 ... $1245.00
Model 1420 ... $945.00 | Model 1520 ... $1495.00

Continental Specialties

Model 3001 Digital Capacitance Meter 275.00
Model 333 Tri-Mode Comparator 295.00
Model LM-3 40-channel Logic Monitor 585.00
Model LM-1 Logic Monitor 60.00
Model LM-2 Logic Monitor 147.00
Model 2001 Sweepable Function Generator 186.00
Model 5001 Universal Counter-Timer 360.00
Model 6001 650 MHz Frequency Counter 385.00
MAX-100 100 MHz Portable Frequency Counter 149.00
PS-500 500 MHz Decade Prescaler 70.00
MAX-50 50 MHz Handheld Frequency Counter 77.00
MAX-550 550 MHz Handheld Frequency Counter 165.00
Model 4001 Pulse Generator 835.00
Model DP-1 Digital Pulser 83.00
Logic Probes ....
Model LP-1 Digital Logic Probe 50.00
Model LP-2 Economy Logic Probe 28.00
Model LP-3 High Speed Logic Probe 77.00
Model LPK-1 Logic Probe Kit 21.95
Logic Probe Accessories 21
Model LTC-1, LTC-2 Logical Analysis Kits ... 220/250

Apple II. 16K or Apple II. Plus $990



16K Apple Upgrade Kit $62.95

Hi-Speed Serial I/O $174.95
Centronics Printer I/O 199.95
Applesoft II Firmware 174.95
Apple Clock 189.95
Introl X-10 System 269.95
Introl X-10 Controller 179.95
Alf Music Synthesizer 259.95
16 Ch. Analog Input 249.95
2 Ch. Analog Output 164.95
13-Key Keypad 119.95
Visi-Calc 149.95
DS65 Digi-Sector $349.00
Apple Graphics Tablet 795.00
D.C. Hayes Modem II 349.95
Disk II w/Controller 475.00
Disk II 575.00
Pascal Lang. System 475.00
Parallel Printer Card 165.00
Communications Card 210.00
Business Software Pkg. 625.00
Corvus 10 Megabyte Drive 4625.00
Integer ROM Card $175.00
Proto Card 21.95
M & R Modulator 29.95
Sanyo Cassette 54.95
16K Upgrade Kit 62.95
Apple Radio Car 159.00
8" Floppy Controller 350.00
Heuristics Speechlab 179.00
Romplus + ... 169.00
Supertalker 279.00
Cashier 250.00

ATARI® 800 & 400
Personal Computer System

ATARI 800 $899.00
ATARI 400 $499.00

ATARI 800 Includes: Computer Console, BASIC Lang. Cartridge, Education System Master Cartridge, BASIC Language Programming Manual, 800 Operator's Manual w/Notebook, Atari 410 Program Recorder, 8K RAM Module, Power Supply, TV Switch Box.

Peripherals
Disc Drive 599.00
Printer 499.00
Program Recorder 69.00

Software – ROM Cartridges
Education System Master Cartridge 28.00
Atari BASIC 55.00
Assembler Debug 55.00
Basketball 42.00
Life 42.00
Super Breakout™ 42.00
Music Composer 55.00
Super Bug™ 42.00
Computer Chess 55.00
Home Finance 55.00

Education System Cassette Programs
U.S. History 35.00
U.S. Government 35.00
Supervisory Skills 35.00
World History (Western) 35.00
Basic Sociology 35.00
Counseling Procedures 35.00
Principles of Accounting 35.00
Physics 35.00
Great Classics (English) 35.00
Business Communications 35.00
Basic Psychology 35.00
Effective Writing 35.00
Auto Mechanics 35.00
Principles of Economics 35.00
Spelling 35.00
Basic Electricity 35.00
Basic Algebra 35.00

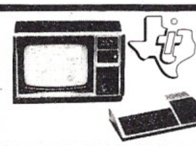
Basic Game and Program Cassettes
Guide to BASIC Programming 18.00
BASIC Game Programs 19.00

Add-On Memory
8K RAM Memory Module 110.00
16K RAM Memory Module 225.00

Diskettes
Blank Diskettes
Disk File Manager 5.00

Accessory Controllers
Driving Controller Pair 18.00
Paddle Controller Pair 18.00
Joystick Controller Pair 18.00

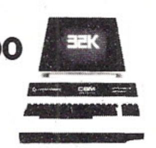
TEXAS INSTRUMENTS INCORPORATED
99/4 PERSONAL COMPUTER



Superior Color, Music, Sound and Graphics – and a Powerful Extended Basic – All Built In.

$1099.00

commodore PET AS LOW AS $775.00



DISCOUNT PRICES

2001-8K 775.00
2001-16KN 975.00
2001-16KB (reg) 975.00
2001-32KN 1250.00
2001-32KB (reg) 1250.00
2022 80-Col Dot Marix Printer 795.00
2023 80-Col Plain Paper Printer 695.00
2040 Dual Mini Disk Drive 1250.00
Pet to IEEE Cable 39.95
IEEE to IEEE Cable 49.95
C2N External Cassette Deck 95.00

CompuCruise
Zemco, Inc.

Model 44 with cruise control **$174.95**

Model 41 without cruise control **$154.95**

An onboard navigational computer for automobiles, trucks, and recreational vehicles. Features cruise control, fuel management system, trip computer, multifunctioned quartz crystal time counter, plus many other functions.

Texas Instruments
LANGUAGE TRANSLATOR THAT SPEAKS

The world's first handheld electronic translator to pronounce and display words, phrases and sentences.

Sugg. Ret. 300.00 ACP Price **$279.95**

LANGUAGE MODULES Sugg. Ret. 60.00
ENGLISH • SPANISH
FRENCH • GERMAN
$54.95 each

HAND HELD COMPUTER SPORTS GAMES
$18.95 EA.

Hand Held While Supply Lasts

ELECTRONIC **FOOTBALL** | ELECTRONIC **SOCCER**

Texas Instruments
ELECTRONIC DIGITAL THERMOSTAT

The Intelligent Way To Cut Energy Costs Year-Round
Use Only The Energy You Really Need – With The Touch Of A Button

The Texas Instruments Electronic Digital Thermostat is the intelligent way to cut energy costs year round. A fully automatic, 24-hour, set and forget comfort system for the home, it can actually pay for itself in less than one season.

Sugg. Reg. 125.00 ACP Price **$104.95**

SINGLE BOARD COMPUTER SELECTION GUIDE

BOARD	PROCESSOR	ACP PRICE	ENCLOSURE
KIM-1	6502	169.00	Add 29.95
SYM-1	6502	249.00	Add 39.95
Cromemco	Z80	409.00	N/A
SD-SBC100	Z80	239.00	N/A
AIM 65	6502	375.00	Add 49.95
Cosmac Vip	1802	199.00	Inc.

RETAIL STORES OPEN MON-SAT
STORE #1 1310 "B" E. Edinger, Santa Ana, CA 92705 — Showrooms, Retail, Warehouse
STORE #2 674 El Camino Real, Tustin, CA 92680 — Specializing in Systems

P.O. Box 17329 Irvine, Calif. 92713
Direct Order Lines: (714) 558-8813
(800) 854-8230 or (800) 854-8241

FOR INTERNATIONAL ORDERS:
1401 E. Borchard (714) 953-0604
Santa Ana, CA 92705 TWX: 910-595-1565

ADVANCED COMPUTER PRODUCTS

GET YOUR 1980 CATALOG

FIRST TO OFFER PRIME PRODUCTS TO THE HOBBYIST AT FAIR PRICES!

1. **Proven Quality** Factory tested products only.
2. **Guaranteed Satisfaction**
3. **Over $1,000,000.00 Inventory**

1980 CATALOG NOW AVAILABLE.

Send $2.00 for your copy of the most complete catalog of computer products. A must for the serious computer user.

STATIC RAM BOARDS

Now less than 1/2¢ per bit!

● S-100 32K (uses 2114)

ASSEMBLED		Kit	
450ns.	499.00	450ns.	469.00
250ns.	539.00	250ns.	499.00

Bare Board 49.95
Bare Board w/all parts less mem. 99.95

● S-100 16K (S-100 Compatible)

WOW!

- Low Power
- 2 MHz or 4 MHz
- 4K Bank Addressable
- Extended Memory Management
- Assembled & Tested

2 MHz **$250.00**
4 MHz **$265.00**

● LOGOS I 8K

ASSEMBLED		KIT	
450 ns.	149.95	450ns.	125.95
250ns.	169.95	250ns.	149.95

Bare PC Board w/Data $21.95
"Special Offer" Buy (4) 8K 450ns. Kits $117.00

The VISTA V-80 Disk Drive System

- 23% more storage capacity than TRS-80
- 120 day warranty
- 40 track patch at NO CHARGE

Single drive system	$ 395.00
Two drive system	$ 770.00
Four drive system	$1450.00
Two drive cable	$ 29.95
Four drive cable	$ 39.95

THE VISTA V-200 FOR EXIDY

Price: Starting as low as **$1199.00**

Item	Storage Capacity	Drive Configuration	Shipping Weight	Price
V200 E-20	400	2 Drive, Single Head	25 lbs.	1199.00
V200 E-22	800	2 Drive, Double Head	25 lbs.	1549.00
V200 E-30	600	3 Drive, Single Head	32lbs.	1524.00
V200 E-32	1.2 MEG	3 Drive, Double Head	32 lbs.	1999.00

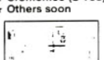
DC-500 HARD DISK INTERFACE

CAMEO DATA SYSTEMS INC

★ Apple II
★ TRS 80
★ Cromemco (S-100)
★ Others soon

HARD DISK SUBSYSTEM I – $6995.00
– Includes CDC Hawk 10 MB Drive
– Interface of Your Choice

DC-500 Controller – TRS 80 Model I, Cables	1500.00
DC-500 Controller – TRS 80 Model II	1500.00
DC-500 Controller – Apple II	1500.00
DC-500 Controller – S-100	1500.00



LOW-COST ADD-ON DISK SUBSYSTEM KIT

★ (2) 8" Drives
★ VISTA Floppy Controller Card
★ Power Supply and Interface Cable
★ CP/M Disk Operating System
★ Box of 10 Diskettes

Low Cost Sys. 1299.00

IMS STATIC RAM BOARDS $699.00

★ Memory Mapping ★ Low Power
★ Phantom ★ Assembled & tested
Recommended by Alphamicrosystems

	250 ns.	450 ns.
8K Static	$209.00	$189.00
16K Static	$449.00	$399.00
32K Static	$729.00	$629.00

ANADEX PRINTER NEW APPLE VERSION

Model DP-8000 compact, impact, parallel or serial. Sprocket feed, 80 cols, 84 lines/min., bi-directional.
New only $875.00
DP-8000AP (for Apple) $875.00

SD SYSTEMS BOARDS

	KIT	ASSEM
SBC 100 Single Board Computer (2MHz)	$265.00	$ 349.00
SBC 200 Single Board Computer (4MHz)	299.00	399.00
Z80 Starter System	319.00	449.00
VDB 8024 Video Display Board	335.00	459.00
Versa-Floppy II	325.00	429.00
Expando PROM	115.00	225.00
SD100 Computer System w/64K	-	6995.00
SD200 Computer System w/64K	-	7995.00

6800 MICROMODULE™ PRICE LIST

MODEL NO.	DESCRIPTION	PRICE
9600A	Single Board Microcomputer	$495.00
9609	Advanced Single Bd. Comp. (6809)	595.00
9601	16 Slot Mother Board	175.00
9602	Card Cage	75.00
9603	8 Slot Mother Board	100.00
9604	Power Supply	275.00
9605	DC Input Power Supply	325.00
9610	Utility Proto Board	39.00
9611	Arith. Proc/Memory Module	495.00
9612	Buffered Utility Proto Board	49.00
9616	32K EPROM/RAM Module	250.00
9617	EPROM Programming Head	250.00
9620	16 Channel Parallel I/O Module	295.00
9622	Serial-Parallel I/O Module	325.00
9627	16K Static RAM Module 470ns	395.00
9629	32K Static RAM 450ns	695.00
9629A	32K Static RAM 200ns	895.00
9630	Card Extender	68.00
9640	Multiple Programmable Timer	395.00
9650	8 Channel Duplex Serial I/O Mod.	395.00
9655	Intelligent Tape Controller	550.00
96103	32/32 I/O Module	275.00
96702	Contact Closure Module	350.00

UNPOPULATED BOARDS (Also Available)

APPLE/EXIDY/EXPANDO TRS 80 16K-UPGRADE KIT

★ Special: TRS80 Schematic .... $ 4.95
★ Expansion Interface Schematic .. $ 4.95
★ Expansion Interface Connector 7.95

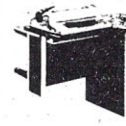
EXPANDORAM II MEMORY KITS

★ Bank Selectable ★ Uses 4116 200 ns.
★ Write Protect ★ Power 8VDC, ±16VDC
★ Phantom ★ Up to 4 MHz
Expando 64 Kit (4116)

16K	$290.00	48K	$444.00
32K	$365.00	64K	$519.00

LETTER QUALITY SELECTRIC!

TRENDATA 1000 COMMUNICATION STATION



Supplied with documentation. Power input: 115VAC ± 10%, 60Hz, 200 Watts. Printer platen, 15" wide. Desk size: 20" x 35" wide x 23" deep. Excellent condition. Guaranteed. Shpg. Wt. 150 lbs.

CAT NO. TDP-RE232	$1399.00
CAT NO. TDP-TTRS80	1399.00
CAT NO. TDP-EBCDIC	1299.00

UV "Eprom" Eraser

Model UVs-11E $69.95
Holds 4 Eprom's at a time. Backed by 45 years experience.
Model S-52T...$265.00
Professional Industrial Model

EMAKO-20.. Reg. $777.00 $599.00

UNBELIEVABLE!!
125 Caps, 60 lpm - Vertical Formal Unit - 96 Characters - Upper/Lower Case - 4.5" to 9.5" Adjustable - 80 col/40 col double width - Full 96 char. ASCII

EMAKO-22............$799.00

Prints a 132 col/line. Available with parallel or serial output at same price.

MIKA 20.............$1280.00

9x7, 125 cps 136 characters/line
Full 15" width. Super for business applications requiring large IBM format paper.

NEW CENTRONICS 737 PRINTER

Uses any paper roll, fanfold, single sheets, 96 character ASCII, 7 x 7 dot matrix, 50 CPS, RS232 or parallel I/O

WOW! $899.00

Z-80/Z-80A/8080 CPU BOARD

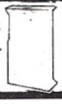
★ On board 2708 ★ 2708 included (450ns.)
★ Power on jump ★ completely socketed
★ Z-80 Assembled and Tested $185.00
★ Z-80 Kit $129.95
★ Z-80 Bare PC Board .... $ 34.95
★ For 4MHz Speed Add $15.00
8080A Kit .... $ 99.95
8080A Assembled $149.95

S-100 MOTHERBOARD SPECIAL

8 slot expandable w/9 conn.
reg $69.95 .... NOW $52.95

SIEMEN'S FLOPPY SALE

- Special buy while supply lasts. — SIEMENS $429.00
- 8" Drive with Double-Density — SHUGART 801R
- 90 Day Warranty — $475.00



ACOUSTIC MODEM

NOVATION CAT™
0-300 Baud
Bell 103
Answer, Originate $169.00

DATA BOOKS ● COMPUTER BOOKS

1980 IC Master	59.95	Intel MCS 80 Manual	7.95
NSC TTL Data	3.95	Intel MCS 40 Manual	4.95
NSC Linear	4.95	AMD 8080A Manual	5.95
NSC Linear App Notes II	3.95	AMD Schottky Databook	4.95
NSC CMOS	3.95	AMI MOS/LSI Data	3.95
NSC Memory	3.95	GI MOS/LSI Data	4.95
Intel Databook	4.95	Harris Analog Databook	4.95
Intel MCS 85 Manual	7.50	TI Linear Control Data	3.95

SALE ● OSBORNE BOOKS ● SALE

	Reg.	Sale
Intro to Micros Vol. 0	~~8.50~~	7.75
Intro to Micros Vol. I	~~8.50~~	7.75
8080A Programming	~~8.50~~	7.75
6800 Programming	~~8.50~~	7.75
Z80 Programming	~~8.50~~	7.75
Vol. II Some Real Microprocessors w/Binder	~~30.00~~	27.50
Vol. III Some Real Support Devices w/Binder	~~20.00~~	18.50
Intro to Micros Vol. III	~~20.00~~	18.50

SALE ● DILITHIUM COMPUTER BOOKS ● SALE

Understanding Computers	~~8.95~~	7.95
8080Microcomputer Experiments	~~13.95~~	11.95
Beginning BASIC	~~9.95~~	8.95
Beginners Glossary & Guide	~~6.95~~	5.95
Peanut Butter & Jelly Guide to Computers	~~7.95~~	6.95
8080 Machine Language Programming	~~7.95~~	6.95
Home Computers Vol. I Hardware	~~7.95~~	6.95
Home Computers Vol. II Software	~~13.95~~	11.95
Starship Simulator	~~7.95~~	6.95

MICROPROCESSORS

Z8001 16 bit to 8Mb	$189.00
Z8002 16 bit to 64K	149.00
Z80	**10.75**
Z80A	14.50
F-8 (3850)	16.95
2650	18.95
CD1802	13.95
8080A	8.50
8080A-4MHz	19.95
SALE 8085	**19.95**
8008-1	14.95
[illegible]	9.90
2901A	14.95
2903 4-bit Superslice	29.95
TMS 9900JL	49.95
CP1600	39.95
6502	11.50
6502A	16.95
IM6100	29.95
6800	**11.75**
6800B 2.0 MHz	19.95
6802P	17.95
8035	19.95
8755	49.95
8748	69.95
6809	64.95
8086	69.95

ADVANCED SUPPORT

AM9511 Arith. Processor	175.00
9512 Arith Processor	175.00
9513 Univ. Timing	79.95
AM9517 DMA Controller	18.95
AM9519 Universal Interrupt	18.95

Z-80 SUPPORT CHIPS

Z80-PIO	2.5 MHz	8.75
Z80A-PIO	4.0 MHz	12.95
Z80-CTC	2.5 MHz	8.75
Z80A-CTC	4.0 MHz	12.95
Z80-DMA	2.5 MHz	29.95
Z80A-DMA	4.0 MHz	36.95
Z80-SIO/0	2.5 MHz	35.95
Z80A-SIO/0	4.0 MHz	39.40
Z80-SIO/1	2.5 MHz	35.95
Z80A-SIO/1	4.0 MHz	39.40
Z80-SIO/2	2.5 MHz	35.95
Z80-SIO/2	4.0 MHz	39.40

8080/8085 SUPPORT

8155/8156 I/O	24.95
8755 I/O with Eprom	64.95
8202 Dyn. Ram Cont.	34.95
8205/74S 138 Decoder	3.95
8212 8 bit I/O	2.75
8214 Priority Int.	5.25
8216 Bus Driver	2.75
8224 Clock Gen.	2.95
8224-4 (4MHz)	9.75
8226 Bus Driver	3.95
8T26 Bus Driver	2.95
8228 Sys. Control	5.50
8238 Sys. Cont.	5.50
8251 Prog. I/O	6.95
8253 Int. Timer	16.95
8255 Prog. I/O	6.50
8257 Prog. DMA	16.95
8259 Prog. Int.	17.95
8275 CRT Controller	59.95
8279 Prog. Keyboard	18.95

6800 SUPPORT CHIPS

6810 128 x 8 Ram	4.75
6820 PIA	5.95
6821 PIA	6.50
6828 Priority Int.	9.95
6834-1 512 x 8 Eprom	16.95
6845/HD46505 CRT Cont.	39.95
6847 Color CRT	49.95
6850 ACIA	5.95
6852 Serial Adapter	5.95
6860 Modem	10.95
6862 Modulator	11.95
6871A 1.0MHz OSC	25.95
6875	8.25
6880 Bus Driver	2.95
MC68488	19.95
68047	24.95

1802 SUPPORT CHIPS

1821 SCD 2K RAM	25.00
1822 SCD 256 x 4 RAM	16.95
1824 CD 32 x 8 RAM	9.95
1852 CD 8 bit I/O	10.95
1854 Uart	10.95
1856 CD I/O	8.95
1857 CD I/O	8.95
1861	12.95

6502 SUPPORT CHIPS

6520 PIA	7.50
6522 Mult.	11.95
6530-002,003,004,005	21.95
6532	19.95
6551	19.95

PROMS

2708 450 ns.	8.25
2708-6 650 ns.	7.50
1702A	4.95
2732	74.95
2716-5V	29.95
2716-5V, 12V	29.95
2758-5V	29.95
5203AQ	13.95
5204AQ	14.95
IM 5610	3.90
SALE 8223 32 x 8	**2.95**
82S115 512 x 8 (TS)	16.95
82S123 32 x 8	4.90
82S126 256 x 4	4.90
82S129 256 x 4 (TS)	4.90
82S130 512 x 4 (OC)	6.50
82S23	6.95
82S131	14.95
82S137	14.95

NOTE: WE PROGRAM PROMS

CHARACTER GEN.

2513-001 (5V) Upper	9.50
2513-005 (5V) Lower	10.95
2513-ADM3 (5V) Lower	14.95
MCM6571	11.75
MCM6571A	11.75
MCM6574	14.50
MCM6575	14.50

UARTS/BAUD RATE

TR1602B (5V, 12V)	3.95
AY51013 (5V, 12V)	4.95
AY51014A/1612 (5-14V)	6.95
AY51015A/1863 (5V)	6.95
TMS 6011 (5V, 12V)	5.50
IM6402	7.95
IM6403	8.95
2350 USRT	9.95
1671B Astros	24.95
SALE TR1472B	**9.95**
MC14411	11.95
4702	14.95
WD1941	9.95
COM 5016	16.95

KEYBOARD ENCODERS

AY5-2376	13.75
AY5-3600	13.75
HD0165	9.95
74C922	9.95
74C923	9.95

STATIC RAMS

	1-24	25-99	100
21L02 450ns	1.30	1.25	.99
21L02 250ns	1.59	1.55	1.15
2111	3.75	3.65	3.55
2112-1	2.95	2.85	2.65
2101-1	2.90	2.70	2.55
2114L-250ns (4045)	8.95	7.50	6.50
2114L-450ns (4045)	**5.50**	**5.25**	**4.75**
4044 250 ns	8.95	7.50	6.90
4044 450ns	**5.95**	**5.95**	**4.75**
EMM4200A	9.75	8.75	7.95
EMM4402	7.95	7.25	6.25
EMM 4804	12.50	11.50	9.95
5101C-E	7.95	7.95	7.25
AMD9140/41	10.95	10.25	9.25
AMD9130/31	12.95	11.95	10.25
1101	1.95	1.75	1.25
P2125/93425 (45ns)	9.95	8.35	8.25
6508 1K x 1 CMOS	7.95	7.95	7.25
6518 1K x 1 CMOS	7.95	7.95	7.25
74S189 64 bit Ram	3.95	3.25	2.50
2147 Low Power 4K Static	19.95	18.95	16.95

DYNAMIC RAMS

416/4116 16K (16 Pin)	8.75
Set of 8 416's	62.95
4115 8K (16 Pin)	6.95
4050 4K x 1 (18 Pin)	4.95
4060 4K x 1 (22 Pin)	4.95
4096 4K x 1 (16 Pin)	3.95
2104 4K x 1 (16 Pin)	4.75
4027 4K x 1 (16 Pin)	4.95

5261	1.95	64K RAMS	1103	1.95
5262	1.95		4008L	4.95
5270	4.95		6605	7.95
5280	4.95		6604	4.95
5290	12.45	$175.00	6002	1.50

CHARGE COUPLED DEVICES

16K CCD - First time offered Fairchild 460 CCD 16K Memory (now you can experiment with CCD technology at a reasonable price. 17 page Application note supplied with each order. Quantity limited)

$18.95 each (reg. 43.00)

SOCKETS

# Pins	Lo-Pro Soldertail	Wire Wrap 3 Level Tin	Gold
8	.15	.32	.45
14	.19	.36	.59
16	.20	.38	.62
18	.24	.59	.84
20	.29	.69	.99
22	.24	.79	1.10
24	.38	.85	1.20
28	.43	1.10	1.49
36	.58	1.25	1.69
40	.60	1.40	1.89

ZERO INSERTION FORCE

16 Pin $5.50 24 Pin $7.50 40 Pin $10.25

LED READOUTS

PART NO	SIZE	COLOR	DESCRIPTION	PRICE
DL704	300	Red	Comm. Cathode	.99
DL707	300	Red	Comm. Anode	.99
FND357	357	Red	Comm. Cathode	.99
FND500	500	Red	Comm. Cathode ±1	.99
[illegible]	500	Red	Comm. Cathode	.99
[illegible]	500	Red	Comm. Anode	.99
FND508	500	Red	Comm. Anode ±1	.99
FND530	500	Green	Comm. Cathode	1.49
FND550	500	Orange	Comm. Cathode	1.49
[illegible]	800	Red	Comm. Cathode	1.75
[illegible]	800	Red	Comm. Anode	1.75
HP5082-7340	600	Red	4 x 7 Hexadecimal	20.95
HP5082-7300	600	Red	4 x 7 Sgl Digit RHD	18.95
HP5082-7731	300	Red	Comm. Anode	.75
TIL305		Red	Array 5 x 7	7.95
TIL308	270	Red	Numerical Display	10.95
TIL309	270	Red	Numerical Display	9.95
TIL311	270	Red	4 x 7 Hex decoding	10.95
MAN2A	320	Red	Array	5.95
MAN104	270	Red	Alpha Numeric	8.95
XAN3061	300	Red	Comm. Anode Right DP	1.25
XAN3062	300	Red	Comm. Anode Left DP	1.25
XAN3063	300	Red	Overflow CA Left DP	1.25
XAN3064	300	Red	Comm. Cathode Right DP	1.25
XAN3051	300	Green	Comm. Anode Right DP	1.99
XAN3052	300	Green	Comm. Anode Left DP	1.99
XAN3053	300	Green	Overflow CA Left DP	1.99
XAN3054	300	Green	Comm. Cathode Right DP	1.99
XAN3081	300	Yellow	Comm. Anode Right DP	2.10
XAN3082	300	Yellow	Comm. Anode Left DP	2.10
XAN3083	300	Yellow	Overflow CA Left DP	2.10
XAN3084	300	Yellow	Comm. Cathode Right DP	2.10

IC SPECIAL PURCHASE

LF 13508 JFET Analog Multi 8 bit	8.95
ICM 7045 Precision StopWatch	23.95
ICM 7205 CMOS LED Stopwatch/Timer	19.95
ICM 7207 Oscillator Controller	7.50
ICM 7208 Seven Decade Counter	18.95
ICM 7209 Clock Generator	6.95
ICL 7107 3½ Digit A/D (LED)	14.95
MC14433P 3½ Digit A/D Converter	13.95
ICL 8211 Voltage Reference	1.95
LM 390N Battery Op. Audio Amp.	3/1.00
LM 1800N PLL FM Sterero	3/1.00
LM 1820N AM Radio	3/1.00
LM 1850N Ground Fault IC	3/1.00
LM 2900N Quad Amplifier	2/1.00
LM 2901N (HM) Quad Comparator	2/1.00
LM 2917N Freq. to Volt Conv.	2/1.99
AY-3-3550 4¾ digit DMM	24.95
AY-5-3507 40MHz DVM	9.95
MEM 4963 Mosfet Smoke Detector	11.95
27S08 32 x 8 Prom (IM 5600)	2.95
G115M 6 channel Mosfet	1.49
ULN 2003A 7 channel Driver	.99
10146 1K x 1 ECL Ram	9.95
MCM 14505 64 bit Static Ram	6.95

7520/25	.50	LM 377	1.25
1488/1489	2/1.99	LM 387	1.25
MK5014 Calc.	3/1.99	LM 3302	3/1.99
74S89	3/1.99	LM 4558	3/1.99
78H05K	6.95	RC 4136	2/1.99
LM323K	4.95	RC 4131	2/1.99
MC1372	8.95	COM 5027	39.95
		COM 5037	49.95

PROCESSOR TECHNOLOGY

NKRA Dynamic Memory Boards, Assembled. Without RAM, PROM, or Data Delay Line.
Up to 64K per Board $139.95

FLOPPY DISKS

CERTIFIED! 40 TRACKS DOUBLE DENSITY

8" 3740 IBM Compat.	$4.39
5¼" Hard 10 Sector	$3.99
5¼" Soft Sector 40 Track	$3.99

Vista computer diskettes by Verbatim. Minimum order 10. For quantity purchases please call.

ALSO AVAILABLE:

DOUBLE SIDED DOUBLE DENSITY

PAK OF 10 5¼"	$56.40
8"	$63.90
Floppy disk hard holes and applicators	CALL

FLOPPY DISK I/O

1771-01 8" & Minifloppy	24.95
uPd372 Nec Floppy	49.95
1781 Dual Floppy	29.95
1791-01 Dual Floppy	36.95
uPd 765 Floppy	49.95

A/D CONVERTERS

8700 8 bit Binary	13.50
8701 10 bit Binary	22.00
8703 8 bit TS	13.50
9400 Volt to Freq. Conv.	7.25
8750 3½ Digit BCD	13.95
1408L6 6 bit	3.95
1408L8 8 bit	5.95
DACOI D to A	5.95

TV CHIPS/SOUND

AY38500-1 6 Games B/W	4.95
AY38515 Color Converter	2.95
AY38603-1 Roadrace Game	8.95
AY38605-1 Warfare Game	CALL
AY38606-1 Wipeout Game	9.50
AY38607-1 Shooting Gallery	8.95
AY38910 Sound Generator	12.95
SN76477 TI Sound Generator	3.95
MM5320/21 TV Synch Gen.	9.95
MM5369 Prescaler	3.95
LM1889 RF Modulator	3.95
MM57100 NSC Color TV	6.95
MM57104 Clock Gen.	3.75
RF Modulator w/Audio	8.95
M&R Modulator	29.95

WAVEFORM GEN.

8038 Function Gen.	3.95
MC4024VCO	2.95
LM566 VCO	1.95
XR2206 Function Generator	5.25

SHIFT REGISTERS

MM500H Dual25	.50
MM5056N Dual 256	2.95
MM5060N Dual 128	2.95
2510A Dual 100	1.95
2847 Quad 80	4.95
3341 Dual 80	4.95
3351 40 x 9 FIFO	17.95
3357 Quad 80	6.95
9403 16 x 4 FIFO	24.95
9408 10 Bit Sequencer	9.95

CTS DIPSWITCHES

CTS206-2	1.75	CTS206-7	1.75
CTS206-4	1.75	CTS206-8	1.95
CTS206-5	1.75	CTS206-9	1.95
CTS206-6	1.75	CTS206-10	1.95

CONNECTORS (GOLD)

DB25P (RS232)	3.25
DB25S Female	3.75
Hood	1.25
Set w/Hood, Sale	$7.50
22/44 W/W, S/T, KIM	2.95
43/86 W/W, S/T, MOT	6.50
50/100 S-100 Connector w/w	4.95
50/100 S-100 Connector s/t	3.95

NAKED PC

Z-80 CPU (Ithaca)	$34.95
8080A CPU	34.95
8K Static RAM (Logos)	24.95
16K Static RAM (2114)	29.95
32K Static RAM (2114)	29.95
Floppy I/O (Tarbell)	39.95
Cassette I/O (Tarbell)	29.95
8K Eprom (2708)	24.95
1702 Eprom Board	30.00
2708/2716 Eprom (Ithaca)	34.95
2708/2716 Eprom (WMC)	30.00
Realtime Clock	34.95
ACP Proto Bd (3M Conn.)	27.95
Vector 8800 Proto	22.20
Vector 8803 11 slot MB	29.95
ACP Extender w/Conn.	18.95
Video Interface (SSM)	31.95
Parallel Interface (SSM)	31.95
13 Slot MotherBoard (WMC)	32.95
9 Slot MotherBoard (WMC)	29.95
8 Slot Mother (expandable)	34.95
Proc-Tech Bare Boards	CALL

COMPUTER SPECIALS

	LIST	SALE
Apple II Plus w/16K	1195.	990.
PET 2001-16N	995.	895.
Exidy Sorcerer w/16K		1099.
Compucolor II w/8K	1895.	1795.
Cromemco Sys III	6990.	6290.
Horizon I w/32K		CALL
TEI PI208 w/32K dual floppy & CRT	4995.	2495.
Pascal Microengine	1995.	1595.
IPSI 1620 Diablo RO	3295.	2695.
Anadex DP 8000	995.	875.
Centronics Micro P-1	595.	395.
Centronics Micro S-1	595.	525.
Soroc IQ 120	995.	795.
Teletype Model 43	1349.	1150.
HiPlot Plotter	1085.	899.
HiPlot Digitizer	795.	735.
Intertube II	895.	784.
Atari 800	999.	899.
Atari 400	549.	499.
TI 99/4	1150.	1099.
Leedex 12" Monitor	149.	139.
Centronics 737	995.	899.
Trencom T-100	375.	349.
Trencom T-200	595.	525.
Sanyo 9" Monitor	220.	169.
Sanyo 15" Monitor	295.	249.
Motorola 22 MHz Monitor	296.	219.

SOCKET SPECIALS

14 Pin w/w-2L	$ .35
16 Pin w/w-2L	.40
18 Pin w/w-2L	.55
20 Pin w/w-2L	.75
40 Pin w/w-2L	1.45
18 Pin s/t	.15
22 Pin s/t	.15

Call for volume pricing.

TERMS: Use check, M/C, VISA, AMEX, CB, or COD. COD requires 25% deposit. Charge Orders please include expiration date. Foreign pay U.S. Funds. Order by phone, mail or TWX. MINIMUM $10.00. Please include magazine issue and phone no. SHIPPING: U.S.A. add $2.00 for first 2 lbs. For surface add 30¢ for additional lbs. For air add 70¢ for additional lbs. FOREIGN: Add 10% shipping and handling. COD's $1.85 extra. Not responsible for typos. Some items subject to prior sale. We reserve right to limit quantities, some items subject to precious metal adder. Retail pricing may vary from Mail Order.

RETAIL STORES OPEN MON-SAT

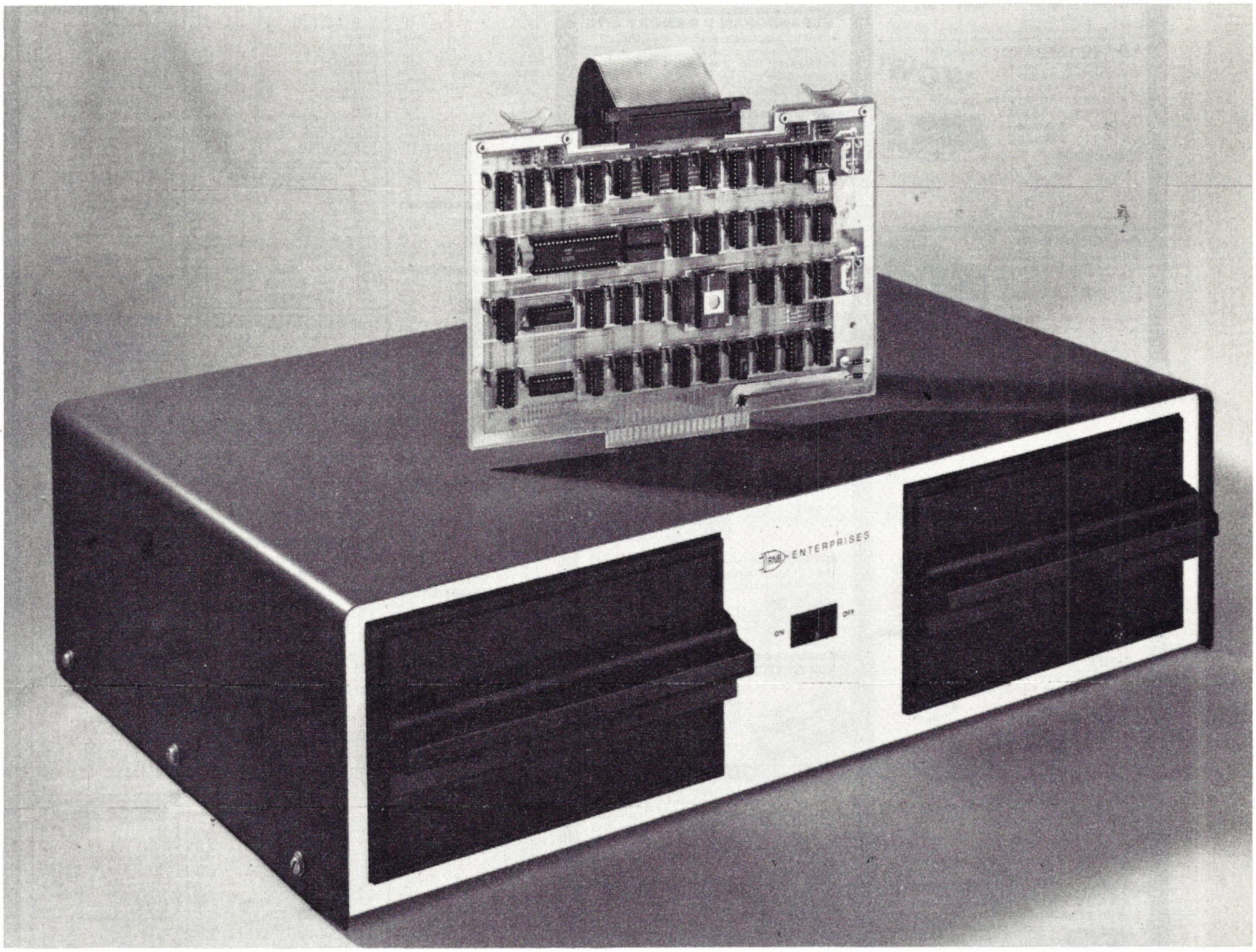
STORE #1 — 1310 "B" E. Edinger, Santa Ana, CA 92705 — Showrooms, Retail, Warehouse
STORE #2 — 674 El Camino Real, Tustin, CA 92680 — Specializing in Systems

P.O. Box 17329 Irvine, Calif. 92713
Direct Order Lines: (714) 558-8813
(800) 854-8230 or (800) 854-8241

FOR INTERNATIONAL ORDERS:
1401 E. Borchard (714) 953-0604
Santa Ana, CA 92705 TWX: 910-595-1565

Reader Service—see page 258

VAK-7 FLOPPY



The VAK-7 Disk System was specifically designed for use with AIM-65, SYM-1, and KIM-1 Microcomputer Systems. The VAK-7 will plug directly into the VAK-1 Motherboard or with the addition of regulators into the KIM-4* Motherboard. The VAK-7 is a complete full size (8") FLOPPY DISK SYSTEM. This system will READ, WRITE, and FORMAT both IBM SINGLE and DUAL DENSITY diskettes. Single-Sided is standard and Dual-Sided is optional. Our Single-Sided drives are set up so they can be converted at a later date to Dual-Sided by the factory, if your storage needs increase.

The VAK-7 system occupies a 4K address space. The system has a 1K block of D.M.A. RAM as a transfer buffer. Also, a 1K block of RAM reserved for D.O.S. pointers, drive status, and catalog information. The remainder of the address is occupied by the resident 2K MINI-DOS. This MINI-DOS is a complete set of subroutines to Read, Write, and Format.

DISK SYSTEM

The MINI-DOS is not a high level Disk Operating System, but contains all the elementary subroutines for implementation of a high level DOS. Since all the functions are in subroutines, the implementation of this system into a dedicated system is simplified.

MINI-DOS SUBROUTINES

Block Move	Read/Write Deleted Data
Seek Track	Format Disk/Test For Bad Sectors
Recalibrate Disk	Initialize Disk
Sense Interrupt Status	Physical Copy (Disk to Disk)
Read/Write Data	Self Test

The VAK-7 is an interrupt driven system, which uses the $\overline{\text{IRQ}}$ vector. Since this is an interrupt driven system, your system processor is only used to move data into or out of the 1K of DMA RAM, issue the command, and check status at the end of the disk operation. Your system processor is free to do other functions, during disk operations because the intelligent disk controller will complete the operation without tying up valuable processor time.

The VAK-7 System comes complete with Disk Controller Board, Interconnecting Cable, a Cabinet with Power Supply (for two Disk Drives) and one Disk Drive. The VAK-7 Controller can handle up to Four Drives.

SPECIFICATIONS:

- Completely assembled, tested, and burned in.
- Occupies address $9000-$9FFF for AIM-65, $9000-$9FFF for SYM-1, or $E000-$EFFF for KIM-1.
- IBM Format; Single Density (128 bytes/sector); Dual Density (256, 512, or 1024 bytes/sector).
- All IC's are in sockets.
- Fully buffered address and data bus.
- Standard KIM-4*BUS (both electrical pin-out and card size).
- Designed for use with a regulated power supply, but has provisions for adding regulators for use with an unregulated power supply.
- Dimensions: Board—10" wide x 7" high (including card-edge)
  Cabinet—23.5" wide x 6.5" high x 16" deep.
- Power Requirements: +5V DC @ 2 Amps.
  117V AC 60Hz @ 2 Amps.

**KIM-4 is a product of MOS Technology/C.B.M.*

PRICE:

Single-drive, 1-sided	$1,299.00
Dual-drive, 1-sided	1,898.00
Single-drive, 2-sided	1,499.00
Dual-drive, 2-sided	2,398.00

Plus Shipping	UPS	Mail (APO, FPO)	International
Single Drive	12.00	32.00	Shipped Air Freight. Freight charges collect.
Dual Drive	16.00	44.00	

For Alaska and Hawaii, use mail rates.

We manufacture a complete line of high quality expansion boards. Use reader service card to be added to our mailing list, or U.S. residents send $1.00 (International send $3.00 U.S.) for airmail delivery of our complete catalog.

RNB ENTERPRISES 52
INCORPORATED
2967 W. Fairmount Avenue
Phoenix AZ 85017
(602)265-7564

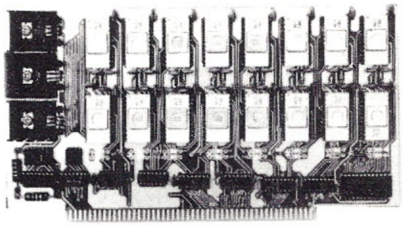
master charge VISA

Reader Service—see page 258

✓Reader Service—see page 258

DIGITAL RESEARCH COMPUTERS
(214) 494-1505

16K EPROM CARD-S 100 BUSS



$59.95 KIT

FIRST TIME OFFERED!
BLANK PC BOARD - $28

USES 2708's!

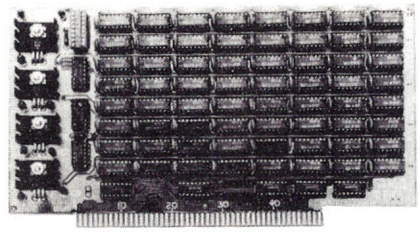
Thousands of personal and business systems around the world use this board with complete satisfaction. Puts 16K of software on line at **ALL TIMES!** Kit features a top quality soldermasked and silk-screened PC board and first run parts and sockets. Any number of EPROM locations may be disabled to avoid any memory conflicts. Fully buffered and has WAIT STATE capabilities.

OUR 450 NS 2708'S ARE $8.95 EA. WITH PURCHASE OF KIT

ASSEMBLED AND FULLY TESTED ADD $30

8K LOW POWER RAM KIT-S 100 BUSS
SALE

PRICE CUT!



$119.50 KIT

21L02 (450 NS RAMS!)

Thousands of computer systems rely on this rugged, work horse, RAM board. Designed for error-free, NO HASSLE, systems use.

ASSEMBLED AND FULLY BURNED IN ADD $35

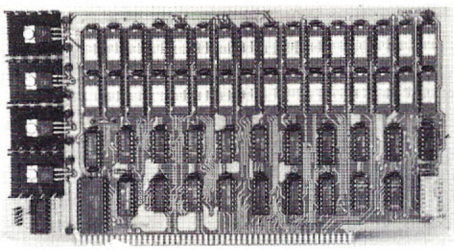
ALL ASSEMBLED BOARDS ARE TESTED AT 4MHZ.

Blank PC Board w/Documentation **$29.95**
Low Profile Socket Set...**13.50**
Support IC's (TTL & Regulators) **$9.75**
Bypass CAP's (Disc & Tantalums) **$4.50**

16K STATIC RAM KIT-S 100 BUSS

PRICE CUT!
$225 KIT

FOR 4MHZ ADD $10



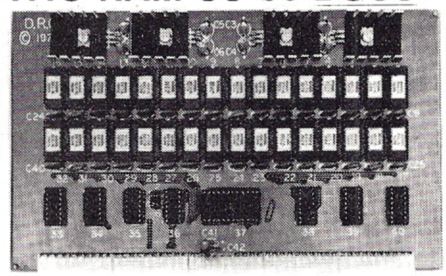
KIT FEATURES:
1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 1.5 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA-**$33**
LOW PROFILE SOCKET SET-**$12**
SUPPORT IC'S & CAPS-**$19.95**
ASSEMBLED & TESTED-**ADD $35**

OUR #1 SELLING RAM BOARD!

16K STATIC RAM SS-50 BUSS

PRICE CUT!



$229 KIT

FULLY STATIC AT DYNAMIC PRICES

FOR SWTPC 6800 BUSS!

ASSEMBLED AND TESTED - $35

KIT FEATURES:
1. Addressable on 16K Boundaries
2. Uses 2114 Static Ram
3. Runs at Full Speed
4. Double sided PC Board. Solder mask and silk screened layout. Gold fingers.
5. All Parts and Sockets included
6. Low Power: Under 1.5 Amps Typical

BLANK PC BOARD—**$26** COMPLETE SOCKET SET—**$12**
SUPPORT IC'S AND CAPS—**$19.95**

NEW! STEREO! NEW!
S-100 SOUND COMPUTER BOARD

At last, an S-100 Board that unleashes the full power of two unbelievable General Instruments AY3-8910 NMOS computer sound IC's. Allows you under total computer control to generate an infinite number of special sound effects for games or any other program. Sounds can be called in BASIC, ASSEMBLY LANGUAGE, etc.

KIT FEATURES:
- ★ TWO GI SOUND COMPUTER IC'S.
- ★ FOUR PARALLEL I/O PORTS ON BOARD.
- ★ USES ON BOARD AUDIO AMPS OR YOUR STEREO.
- ★ ON BOARD PROTO TYPING AREA.
- ★ ALL SOCKETS, PARTS AND HARDWARE ARE INCLUDED.
- ★ PC BOARD IS SOLDERMASKED, SILK SCREENED, WITH GOLD CONTACTS.
- ★ EASY, QUICK, AND FUN TO BUILD, WITH FULL INSTRUCTIONS.
- ★ USES PROGRAMMED I/O FOR MAXIMUM SYSTEM FLEXIBILITY.

Both Basic and Assembly Language Programming examples are included.

COMPLETE KIT! $84.95 (WITH DATA MANUAL)

BLANK PC BOARD W/DATA $31

SOFTWARE:

SCL Interpreter coming soon! Our new Sound Command Language interpreter along with the Register Examine/Modify (REM) routines and Sound Effects Library (SEL) will be available soon in EPROM. SCL makes sound effects programming generally easier and quicker than that written in Basic or Assembly Language. An SCL users group will be formed, and the best new SCL programs submitted will be added to the Sound Effects Library in EPROM.

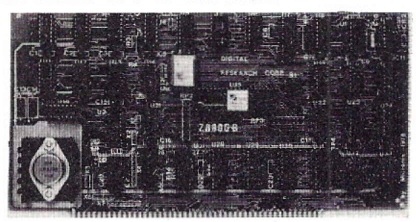
S-100 Z80 CPU CARD

Perfect for OEM's

$169.95

WIRED! NOT A KIT!

4 MHZ



ASSEMBLED AND TESTED! READY TO USE! Over 3 years of design efforts were required to produce a **TRUE** S-100 Z80 CPU at a genuinely bargain price!

BRAND NEW!

FEATURES:
★ 2 or 4 MHZ Operation. ★ Generates MWRITE, so no front panel required.
★ Jump on reset capability. ★ 8080 Signals emulated for S-100 compatability.
★ Top Quality PCB. Silk Screened. Solder Masked. Gold Plated Contact Fingers.

SALE! LOW POWER - 300NS 2114 RAM SALE!

FOR 4MHZ **8 FOR $44**

4K STATIC RAM'S. MAJOR BRAND, NEW PARTS.
These are the most sought after 2114's, LOW POWER and 300NS FAST.
8 FOR $44

NEW! G.I. COMPUTER SOUND CHIP

AY3-8910. As featured in July, 1979 BYTE! A fantastically powerful Sound & Music Generator. Perfect for use with any 8 Bit Microprocessor. Contains: 3 Tone Channels, Noise Generator, 3 Channels of Amplitude Control. 16 Bit Envelope Period Control, 2-8 Bit Parallel I/O. 3 D to A Converters, plus much more! All in one 40 Pin DIP. Super easy to interface to the S-100 or other busses.

SPECIAL OFFER: $14.95 each Add $3 for 60 page Data Manual.

Digital Research Computers
(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 494-1505

TERMS: Add $1.00 postage, we pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. 90 Day Money Back Guarantee on all items. Orders over $50. add 85¢ for insurance.

NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

The Place To Buy Computers

CROMEMCO Z-2H
HARD DISK

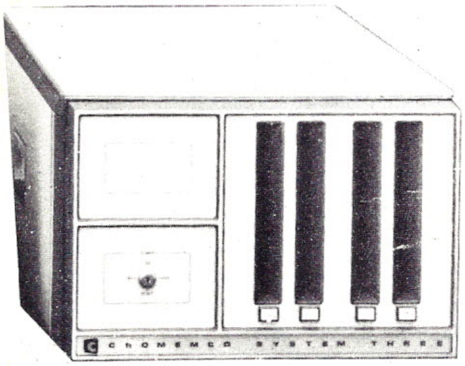
• Full 11-megabyte hard disk system • fast Z80A 4 MHz processor • two floppy disk drives • 64K RAM memory • RS232 special interface • printer interface • extensive software available

List $9995 . . . OUR PRICE ONLY **$8489**

CROMEMCO HDD

11/22 megabyte hard disk for use with existing systems. DMA controller, transfer rate of 5.6 megabytes/sec.

HDD-11, List $6995 . . OUR PRICE $ 5,939
HDD-22, List $11,995 10,189



CROMEMCO SYSTEM 3

— NOW DOUBLE THE CAPACITY! —

Features 4 MHz CPU, 64K of RAM, dual-sided PerSci 299B floppy disk drive (provision for installing a second 299B), RS232C interface, printer interface. All Cromemco systems are assembled and tested, ready to use.

With 64K of RAM, List $6990

OUR PRICE ONLY $5890

CROMEMCO SYSTEM 2

With 64K of RAM, two minifloppy disk drives, RS232 interface and printer interface board.

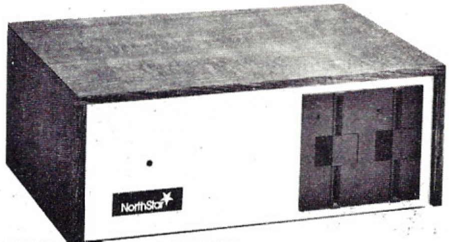
System 2 w/64K RAM, List $3990 . . . . . . $3390

CROMEMCO Z-2

Can be rack mounted. Z-80 processor, 21 slots, power supply, front cover panel. Includes fan and all edge connectors. Assembled and tested.

Z-2W, Assem., List $995 . . . . $845

NORTH STAR HORIZON



HORIZON 1 KITS —

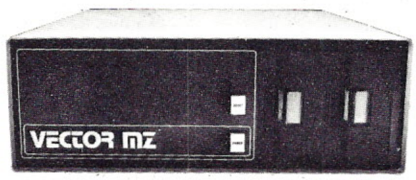
16K, Double Density, List $1599 . . $1474
32K, Double Density, List $1849 . . 1684
32K, Quad Density, List $2049 . . . 1869

HORIZON 2 KITS —

16K, Double Density, List $1999 . . $1824
32K, Double Density, List $2249 . . 2034
32K, Quad Density, List $2629 . . . 2359

Kits incl. 12 edge connectors, 2 serial ports, parallel port and extra drive cable. (Subject to availability and price changes.) Call or write for current prices on assembled units.

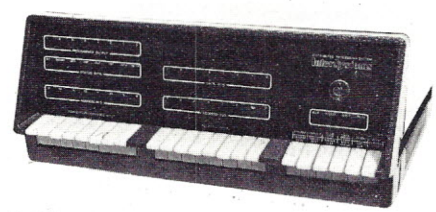
VECTOR MZ



Now 64K with Bank Select — Complete Peachtree Business Software Package also available — Call for details.

INTER SYSTEMS

(formerly ITHACA AUDIO)



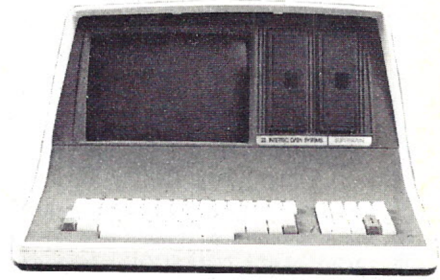
S-100 mainframe, features a 4 MHz Z-80A CPU and a full feature front panel. 20-slot actively terminated motherboard, with 25-amp power supply (50/60 Hz operation, incl. 68 cfm fan).

DPS-1, List $1145 . . . . $1029

SD SYSTEMS

SDS-100, List $7995 . . . . . ONLY $6795
SDS-200, List $8995 7645

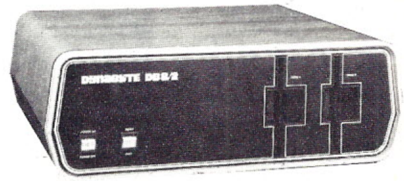
SUPERBRAIN®
By INTERTEC



Available with 32K, 48K & 64K

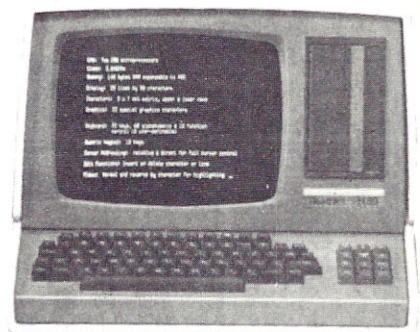
Totally self-contained in a single box; 32K, 48K, or 64K Version; Uses two Z-80 CPU's; Commercial-type terminal with 12" monitor (like the Intertube); Dual double-density minifloppies w/360 Kilobytes of storage capacity; I/O ports included; Expandable (if needed) with an external S-100 bus interface; Comes with CP/M™ operating system; extensive software support; with 32K of RAM, List $2995 . . . . ONLY $2685

DYNABYTE



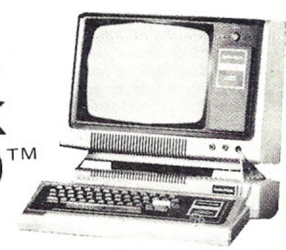
48K and 64K models, single and double density, dual mini disk (77 track), standard 8" and dual-sided 8" systems. Call or write for prices.

HEATH



WH-89 — All-in-one computer. Features two Z-80's, 16K to 48K. Call or write for prices.

RADIO SHACK TRS-80™



10% OFF

WRITE FOR FREE CATALOG

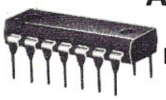
Above prices reflect a 2% cash discount (prepaid prior to shipment). Add 2% to these prices for credit cards, C.O.D., etc. Prices are f.o.b. shipping point. Prices are subject to change and offers subject to withdrawal without notice.

MiniMicroMart, Inc.

✓275 1618 James Street, Syracuse NY 13203 (315) 422-4467 TWX 710-541-0431

Make The Shack® Your Parts Place!
Low Prices and New Items Every Day

100% Prime CMOS And TTL ICs



Low As **49¢** Each

CMOS

Type	Cat. No.	ONLY
4001	276-2401	.69
4011	276-2411	.69
4012	276-2412	.79
4013	276-2413	.99
4017	276-2417	1.69
4020	276-2420	1.69
4021	276-2421	1.69
4023	276-2423	.69
4027	276-2427	.99
4028	276-2428	1.29
4046	276-2446	1.89
4049	276-2449	.79
4050	276-2450	.79
4066	276-2466	1.89
4070	276-2470	.79
4511	276-2447	1.69
4543	276-2491	1.99

TTL

Type	Cat. No.	ONLY
7400	276-1801	.49
7402	276-1811	.49
7404	276-1802	.49
7441	276-1804	.99
7447	276-1805	.99
7448	276-1816	.99
7473	276-1803	.49
7475	276-1806	.89
7476	276-1813	.59
7490	276-1808	.89
7492	276-1819	.89
74154	276-1834	1.39
74192	276-1831	1.19
74193	276-1820	1.19

Top Octave Synthesizer IC 6.99

S50240. Ideal for electronic keyboard instruments! Generates 13 equally timbred notes, C_8 to $C_8{}^{\#}$. CMOS-compatible square wave outputs. 11 to 16VDC, single supply. 16-pin DIP. With data. **276-1780** **6.99**

Rhythm Pattern Generator IC 5.99

MM5871. Build your own rhythm box or drum synthesizer! 6 patterns can be used alone or in any combination. Trigger outputs for wood block, brushes, bongo, snare and bass drums. With data and applications circuits. 16-pin DIP. **276-1785** **5.99**

Why Solder? No-Strip Wire Wrapping Tool Speeds Up Wiring

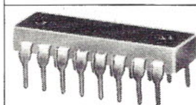
12.95

For Fast, Pro-Quality Wiring

Quick-Wrap Tool.™ Ideal for daisy chain or point-to-point wiring. Cuts and strips wire. Easy to load. For 0.025" square posts. **276-1572** **12.95**

30-Gauge Kynar Wire. For above. 50' spool. **Red, 278-501. White, 278-502. Blue, 278-503** . . . . **Each 1.99**

2102L and 2114L Static RAMs



Low As **2.49** 2102L **Buy 8 2102Ls Save $5**

Type 2102L. 100% prime 1024 x 1 NMOS RAM. Single 5V supply. 16-pin DIP. Under 450ns. Buy 8 and save! **276-2501** . . . . **2.49 Each 8 for 14.95**

Type 2114L. 100% prime 1024 x 4 NMOS RAM. 18-pin DIP. Under 450ns. Single 5V supply. **276-2504** **10.95**

3½-Digit LCD Readout Multitester

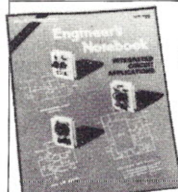
79.95

- *20 Ranges*
- *10 Megohm Input*
- *Big 0.4" Display*

Liquid crystal display is easy to read, even in sunlight. Automatic polarity and zero adjust. Over-range and low battery indicators. Voltage: 500V max. AC, 1000V max. DC. AC/DC Current: 2-20-200 mA. Resistance: 200-2k-20k-200k-2M-20MΩ. With test leads, instruction manual. 1¾x3¾x6¾." Requires 9V battery or adapter. **22-198** **79.95**

AC Adapter for above. 273-1431 . . . . **4.95**

Engineer's Notebook of IC Circuits

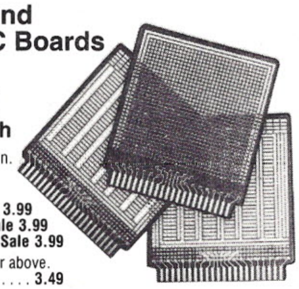


1.99

Sold Only at Radio Shack

You'll constantly use this "must-have" sourcebook. Applications and circuit examples for most popular linear and digital ICs, truth tables, helpful tips, more. All circuits have been built and tested. 128 pages. **276-5001** . . . . **Only 1.99**

Multipurpose, Digital and Op-Amp Edge-Card PC Boards



Save 20% Reg. 4.99 **3.99 Each**

Top quality! Low loss epoxy glass construction. All are 4½x4," with ¹⁄₁₆" grid and dual 22-contact edge. 1368 holes.

Multipurpose. 276-155 . . . . **Sale 3.99**
2-Voltage (Digital). 276-156 . . . . **Sale 3.99**
3-Voltage (Op-Amp). 276-157 . . . . **Sale 3.99**

22-Pin Dual-Edge Wire Wrapping Socket. For above. **276-1550** **3.49**

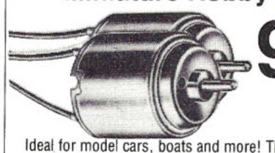
NEW! Project Case With Panel

- *Flip Open Cover*
- *High Impact Plastic*

2.99

This deluxe molded case is ideal for many projects. Easy-to-work reversible inner panel accepts switches, meter or readouts. Hinged lid has snap-lock closure. 5½x3⅜x2." **270-290** **2.99**

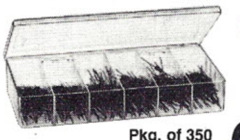
Miniature Hobby Motors



99¢ Pkg. of 2

Ideal for model cars, boats and more! These PM-type motors operate with 1½ to 8VDC and deliver up to 10,000 RPM at no load. Motor: 1½x⅞." Shaft: ⁵⁄₆₄x¼." Long leads for easy hookup. Be creative at a sensible low price! **273-206** . . . . **2 for 99¢**

SALE! Resistor Packs



Save 30% Reg. 9.95

Pkg. of 350 **6.95 Each**

Storage Box Included!

Less than 2¢ each! Precision-made resistors, popular values: 10Ω to 10 megs. Stock up!

½-Watt. 10% tol. 47 or more popular values. **271-601** . . . . **Pkg. of 350**/(Reg. 9.95) **Sale 6.95**

¼-Watt. 5% tol. 56 or more popular values. **271-602** . . . . **Pkg. of 350**/(Reg. 9.95) **Sale 6.95**

NEW! 12-Volt Lamp Assemblies

Built-In Long Life Bulb

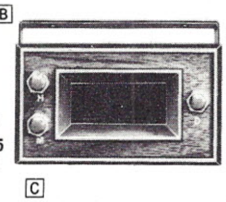
1.89 Pkg. of 2

One red, one green. Complements any project where high visibility is needed. Simple one-piece construction with snap-in mounting fits ½" diameter hole. 4" wire leads. **272-332** . . . . **Pkg. of 2/1.89**

SALE! Clock Cases & Bezel

Save Up to 33%

A



B

C

A **Digital Display Case.** Ideal for clocks, counters, testers. Mounts up to four 0.6" or eight 0.3" readouts. **270-285.** Reg. 3.95 . . . . **Sale 2.95**

B **Auto Project Case.** Mounts in or on dash. Companion enclosure to MA1003 Auto Clock Module. Accepts 3 push switches (not included). Blue lens. 3½x2⅛x2." **270-303.** Reg. 5.95 . . . . **Sale 3.95**

C **Bezel With Lens.** High contrast lens, ideal for LED displays. 3⁹⁄₁₆x1" lens opening. With hardware. **270-301.** Reg. 3.95 . . . . **Sale 2.95**

WHY WAIT FOR MAIL ORDER DELIVERY? IN STOCK NOW AT OUR STORE NEAR YOU!

Radio Shack®

A DIVISION OF TANDY CORPORATION • FORT WORTH, TEXAS 76102
OVER 7000 LOCATIONS IN 40 COUNTRIES

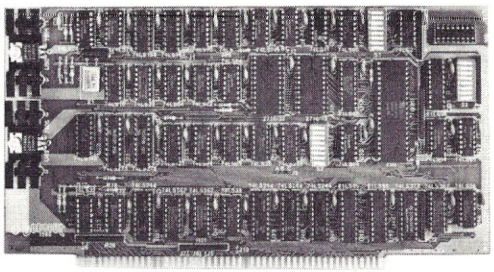
Prices may vary at individual stores and dealers

the FIRST of the New Generation

Our innovative **Z-80A CPU board** is truly the first of a new generation of S-100 bus equipment . . . a generation that's designed to accomodate multi-user setups and other high level industrial, scientific, and commercial applications.

This CPU card contains all the good Z-80 features of other boards, but also features full compliance with the proposed IEEE S-100 bus standards, provision for adding two EROMs for 4K to 8K of on-board ROM (2716 or 2732 – not included with board), power on jump to any of 256 boundaries, on-board fully maskable interrupts at port FE (hex) for interrupt driven systems, 2 or 4 MHz operation, power on clear that generates preset and slave clear according to IEEE specs, selectable automatic wait state insertion for servicing M1* instructions – MRQ* – I/O RQ* – or the on-board ROM (individually or severally selectable), non-maskable interrupt on bus pin 12 as per IEEE specs, and we've also included on-board IEEE compatible extended addressing at port FD (hex).

These advanced features give you the power you need for future expansion, as well as the system flexibility that comes from superior design . . . but perhaps best of all, the price is competitive with boards that do a whole lot less. **The Z-80A CPU board is available for $225 unkit, $295 assembled, and $395 for units qualified under our Certified System Component program.**



Next month: The second new generation CPU board, featuring our 8085/8088 dual processing technique, as well as the Spectrum color graphics board. Please do **not** call for information on these products; send an SASE and we'll send the information to you.

HIGH PERFORMANCE S-100 MOTHERBOARDS

6 slot: $89 unkit, $129 assm

12 slot: $129 unkit, $169 assm **19 slot: $174 unkit, $214 assm**

Ideal for use with the above enclosure. **Unkits** have edge connectors and termination resistors pre-soldered in place for easy assembly. Meets or exceeds IEEE S-100 specs; includes true active termination, grounded Faraday shield between all bus signal lines, and edge connectors for all slots.

LOOKING FOR MEMORY?

Our boards are static, run up to 5 MHz, meet the IEEE S-100 standards, low in power, and include a 1 year warranty. Choose from **unkit** (sockets, bypass caps pre-soldered in place), **assembled**, and boards qualified under our high-reliability **Certified System Component** (CSC) program.

Memory Name	Buss & Notes	Unkit	Assm	CSC
8K Econoram* IIA	S-100	**$169**	**$189**	**$239**
16K Econoram XIV	S-100 (1)	**$299**	**$349**	**$429**
16K Econoram X-16	S-100	**$329**	**$379**	**$479**
16K Econoram XIIIA-16	S-100 (2)	**$349**	**$419**	**$519**
16K Econoram XV-16	H8 (3)	**$339**	**$399**	n/a
16K Econoram IX-16	Dig Grp	**$319**	**$379**	n/a
24K Econoram VIIA-24	S-100	**$449**	**$499**	**$599**
24K Econoram XIIIA-24	S-100 (2)	**$479**	**$539**	**$649**
32K Econoram X-32	S-100	**$599**	**$689**	**$789**
32K Econoram XIIIA-32	S-100 (2)	**$649**	**$729**	**$849**
32K Econoram XV-32	H8 (3)	**$649**	**$749**	n/a
32K Econoram IX-32	Dig Grp	**$599**	**$679**	n/a
32K Econoram XI	SBC/BLC	n/a	n/a	**$1050**

Econoram is a trademark of **Godbout Electronics.**

(1) Extended addressing (24 address lines). Addressable on 4K boundaries.
(2) Compatible with all bank select systems (Cromemco, Alpha Micro, Etc.); addressable on 4K boundaries.
(3) Bank select option for implementing memory systems greater than 64K.

THE GODBOUT COMPUTER BOX

$289 desktop, $329 rack mount

Quiet fan, dual AC outlets and fusehold, heavy-duty line filter, black anodized front panel, and card guide. Ask about our matching power supply.

S-100 2708 EROM BOARD $85 unkit

4 independently addressable 4K blocks. Includes all support chips and manual, but does not include 2708 EROMs.

S-100 ACTIVE TERMINATOR BOARD $34.50 kit

Plugs into older, unterminated motherboards to improve performance.

S-100 MEMORY MANAGER BOARD

$59 unkit, $85 assm, $100 CSC

Adds bank select and extended addressing to older S-100 machines to dramatically increase the available memory space.

2S "Interfacer I" S-100 I/O BOARD

$199 unkit, $249 assm, $324 CSC

Dual RS-232 ports with full handshake. On-board crystal timebase, hardware UARTS, much more.

3P PLUS S "INTERFACER II" I/O BOARD

$199 unkit, $249 assm, $324 CSC

Includes 1 channel of serial I/O (RS-232 with full handshake), along with 3 full duplex parallel ports plus a separate status port.

PASCAL/M™ + MEMORY SPECIAL

PASCAL can give a microcomputer with CP/M more power than many minis. You can buy our totally standard **Wirth PASCAL/M™ 8" diskette**, with manual and Wirth's definitive book on PASCAL, **for $150 with the purchase of any memory board**. Specify Z-80 or 8080/8085 version. PASCAL/M™ available separately for $350.

SEE COMPUPRO PRODUCTS IN PERSON: Many of these products are stocked by finer computer stores world-wide, or write us if there's no dealer in your area.

TERMS: Cal res add tax. Allow 5% for shipping, excess refunded. VISA*/Mastercharge* call our 24 hour order desk at **(415) 562-0636**. COD OK with street address for UPS. Sale prices good through cover month of magazine; other prices are subject to change without notice.

CompuPro™ from GODBOUT ELECTRONICS ✓42

Bldg. 725, Oakland Airport, CA 94614 (415) 562-0636

10-DAY FREE TRIAL

SEND FOR OUR FREE CATALOG

Lear Siegler Terminals & Printers at Tremendous Savings

ADM-3A. Industries' favorite dumb terminal for some very smart reasons. 12" diagonal screen • Full or half duplex operation at 11 selectable data rates • 1,920 easy-to-read characters in 24 rows of 80 letters • RS-232C interface extension port • Direct cursor addressing.
Reg. Price. $895 .... **Sale! $795**

ADM-31. A terminal that's too smart to be considered dumb. Comes complete with keyboard, control logic, character generator, refresh memory & interface • Displays two pages of text instead of one • Field protect mode • Factory installed selected parameters.
Reg. Price. $1,450 .... **Sale! $1,195**

ADM-42 with Keyboard. Reg. Price. $1,795 **Sale! $1,595**
ADM-42 without keyboard. Reg. Price. $1,595 .... **Sale! $1,450**

Model 310 Ballistic Printer (Serial/Parallel)
Reg. Price. $2,045 .... **Sale $1,989**

300 Series Ballistic Printers. This application-oriented matrix printer is built to last. Features include: Ballistic Printing • Positive 180 cps • Bi-directional Printing • A character buffer that is optionally expandable to 2,048 characters • 512 character buffer standard.

ADM-42. The semi-intelligent terminal that provides you with flexibility of format, security, editing, interface, and transmission. Two-page display standard (Optionally expandable to eight) • Blanking, blinking, and reverse fields • Three ways to TAB • 16 Function keys do the work of 32 • Detachable keyboard.

We Stock Lear Siegler Accessories — Call For Details.

The remarkable TI-99/4 Home Computer. Compare it. Dollar for dollar. Feature for feature.

Superior color, music, sound & graphics - and a powerful extended BASIC - all built in. Plus a uniqe new Solid State Speech™ Synthesizer and TI's special Solid State Software.™

Comes complete with the Features you want:
- **Powerful TI-BASIC**
- **Up to 72K total memory capability - 16K RAM, 26K ROM plus up to 30K ROM in TI's Solid State Software Command Modules.**
- **16 - color graphics capability**
- **Music & sound effects**
- **Built-in equation calculator**
- **High quality 13" color monitor**

Call our Sales Dept. for Complete Description & Specs.
TI-99/4 Home Computer w/Monitor .... **$1,050.**
TI-99/4 Home Computer w/o Monitor .... **$8.95.**
TI User Reference Manual .... **$9.95.**

TI CALCULATORS - Three of the finest from the first.
TI Programmable 59 .... **$245.00**
TI-58C Programmable Calculator (w/continuous memory) **$104.00**
TI Programmer **$59.00**
Add convenient versatile printing capabilities to your TI Programmable 58C or 59 calculator with the PC-100C thermal printer, plotter.
TI PC-100C **168.00**
TI Talking Translator. The translator that actually speaks! $300.00 Language Modules for the TI Translator $49.95 Ea. Available in : Spanish, French, English & German.
SPEAK & SPELL™ - TI's remarkable learning aid with electronic voice & brain. &69.95 Modules for Speak & Spell $17.95 Ea: Super Stumpers (grades 4-6), Vowel Power (Ages 7 & up), Super Stumpers (Grades 7 & 8)

ADDITIONAL TI-99/4 ACCESSORIES

Speech Editor	**$99.95**
Demonstration	**$69.95**
Diagnostic	**$29.95**
Joysticks	**$29.95**
Speech Box	**$149.95**
Dual Cassette Cable	**$19.95**
Math Dictionary	**$29.95**
Beginners Basic	**$9.95**

SOLID STATE SOFTWARE COMMAND MODULES

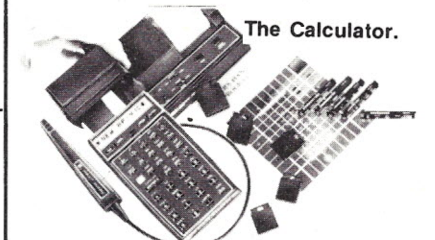
Entertainment:	
• **Football**	**$29.95**
• **Video Chess**	**$69.95**
• **Video Graphs**	**$19.95**
Home Management/Personal Finance:	
• **Home Financial Decisions**	**$22.95**
• **Household Budget Mgt.**	**$44.95**
Education:	
• **Early Learning**	**$29.95**
• **Physical Fitness**	**$29.95**
• **Number Magic**	**$19.95**
• **Beginning Grammar**	**$29.95**

A Calculator, A system, A Whole New Standard.

The finest calculators available for Science, Engineering, & Business.

HEWLETT-PACKARD'S HP-41C

The Calculator.

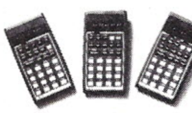


Features over 130 functions and offers up to 400 lines of program memory or 63 data storage registers — expandable to 319 registers or up to 2,000 lines. RPN Logic. Alpha-numeric capabilities let you communicate with the calculator in English. Customization features allow you to totally reassign the keyboard functions. Continuous memory....... **HP-41C Calculator $288.00**

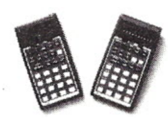
The System.

Memory Modules. For storing programs or up to 2,000 lines of program memory. **$45.00**
"Extra Smart" Card Reader. Records programs and data back onto blank mag-cards. **$179.00**
The Printer. Upper and Lower case. High resolution plotting. Portable. Thermal operation. **$320.00**
Application Modules.................. **$45.00 EACH.**
Standard pac, Statistics, Math, Financial, & Surveying.

HP SERIES E CALCULATORS



HP-31E — Scientific. Trigonometric, exponential & math functions. **$49.95**
HP-32E — Advanced Scientific with Statistics. All HP-31E functions plus hyperbolies and comprehensive statistics. **$66.95**
HP-33E — Programmable Scientific. A programmable science, math and statistical calculator. **$79.95**
Also available with continuous memory.... **86.40**



HP-37E — Business Management. Best choice for a business/finance calculator........ **$69.95**

HP-38E — Advanced Financial with Programmability. All the features of the HP-37E plus a lot more power. **$108.00**
Also available with continuous memory....... **$144.95**
THE ALL NEW HP-34C. Advanced Continuous Memory Scientific Programmable with an impressive array of programming features. **$144.95**

CompuMart STOCKS THE COMPLETE LINE OF MATROX PRODUCTS. CALL FOR SPECS.

Based on our policy of offering our customers only the finest in microcomputers, CompuMart is pleased to announce that we now carry the new generation of Personal Computers by Atari.™

ATARI

INTRODUCTORY SPECIALS

To celebrate our commitment to Atari, we are offering the following Atari Specials:
(1) **Buy additional memory for your computer, 8K or 16K, and we will double the amount of memory FREE!**
(A potential savings of $200.)
(2) **Buy the Atari 800 Computer and take $100 off the purchase price of the Atari 810 Disk Drive or the Atari 820 printer.**

ATARI® 800™
PERSONAL COMPUTER SYSTEM

Comes with:
- **Computer Console**
- **BASIC Language Cartridge**
- **Education System Master Cartridge**
- **BASIC Language Programming Manual**
- **800 Operator's Manual**
- **ATARI 410 Program Recorder**
- **Invitation to Programming™ Cassette**
- **8K RAM Memory Module™**
- **10K ROM Operating System**
- **Power Supply**
- **TV Switch Box**

SPECIFICATIONS:
High resolution color graphics
57 key full stroke keyboard
Built-in RF modulator for channel 2/3 operation with standard TV set
Composite video output for use with monitor
Internal Speaker
Two cartridge slots for rapid program insertion
Four internal slots for expansion up to 48K RAM
6502B Microprocessor
High speed serial 1/0 port

Atari 800 Computer System $995.95

ATARI® 820™ PRINTER
High resolution dot matrix impact printer
Uses standard ⅜ inch roll paper and ribbon
40 characters per line
Speed: 40 characters per second
UL approved

Atari 820 Printer $599.95

ATARI® 810™ PRINTER
Uses standard 5¼ inch diskettes
88K bytes storage per diskette
Up to four disk drive units can operate with the system
Average data access time: 236 milliseconds
Power: AC adapter; UL approved

Atari 810 Disk Drive $699.95

Upgrade your computer with additional memory. (Note that the Atari 800 Computer comes with 8K of RAM memory and will accept up to 48K.)

Atari 8K RAM Memory Module $124.95
Atari 16K RAM Memory Module $199.95

CompuMart ✓43
Dept. KB5,
270 Third St.,
Cambridge, Mass.
02142
TO ORDER CALL:
1-800-343-5504
IN MASS. CALL: (617) 491-2700

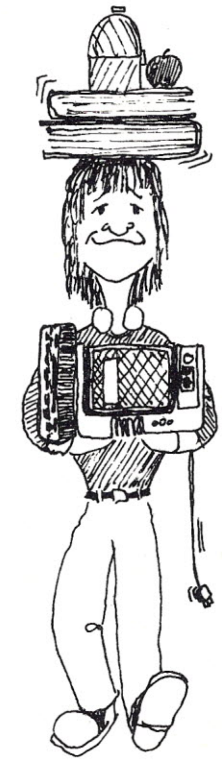
IMPORTANT ORDERING INFORMATION
All orders must include 4% shipping and handling. Mass. residents add 5% sales tax; Mich. residents 4%. for sales tax.

Phones open from 8:30 a.m. to 5:30 p.m. EST, Mon.-Fri. • P.O.'s accepted from D&B rated companies—shipment contingent upon receipt of signed purchase order • All prices are subject to change without notice • Most items in stock for immediate shipment—call for delivery quotation • In the Ann Arbor area? Our retail store is open 11:00 a.m. to 7:00 p.m. Tues.-Fri., 10:00 a.m. to 5:00 p.m. Saturdays (closed Sun. and Mon.)

****IF NOT SATISFIED RETURN PURCHASE WITHIN 10 DAYS FOR A FULL REFUND.***

Featured Next Month—Computers in the Classroom

Don't Miss It!



In June, school is out for the summer, and teachers have to start preparing for the next academic year. The June issue of *Kilobaud Microcomputing* will feature a special section on microcomputers in education, written by educators and computerists. Articles and programs will provide overviews and specific applications for educators already using microcomputers as educational tools, or for those contemplating introducing microcomputers into the curriculum.

Topics will include:

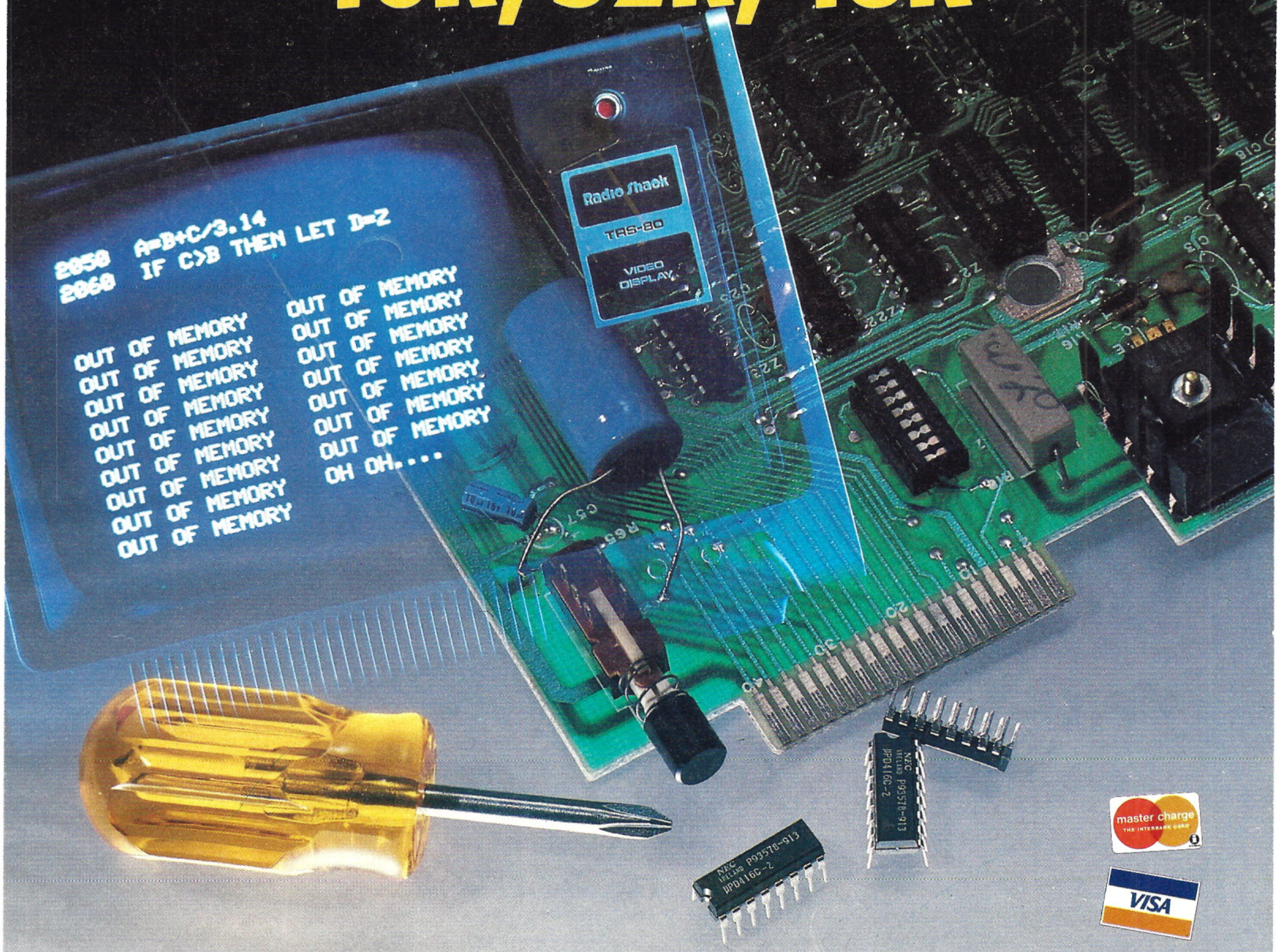
- *math and science programs*
- *a computerized gradebook*
- *a computer survival course for kids*
- *a look at one school's attempt to bring microcomputers into the classroom*
- *plus other articles relating to the growing use of microcomputers in education*

Whether you're actively involved in the teaching discipline or simply support your school system with your tax dollars, you won't want to be truant when *Microcomputing* takes you to school next month. It promises to be a learning experience for everyone.

MICROCOMPUTING T.M. LIST OF ADVERTISERS

***This advertiser prefers to be contacted directly.**

Memory Shortage Cures: 16K, 32K, 48K



Dealer Inquiries Invited

The Product. Only high quality, prime, burned-in and tested 4116 16K dynamic RAMs. Don't be caught unaware! All TRS-80 memory expansion kits are not the same. UHF Associates' memory expansion gives you high quality coupled with outstanding performance. And with their fast 200 NS minimum access time (less CPU wait states) UHF's 4116 16K dynamic RAMs provide both storage and speed that won't disappoint you later down the road.

The Price. 16K Memory Expansion Kit for either computer (pre-programmed DIP shunts included) or expansion interface, $95. More? 32K Kit for expansion interface, $180. Most? 48K Kit for computer and expansion interface, $265.

The Promise. "Thou shalt not wait, worry or fret." You'll get immediate post-paid delivery from in-stock inventory. You'll get a full 12 month warranty. That's about four times the warranty others offer. And for installation, you'll get UHF's "goof-proof" instructions. All you'll need is a screwdriver and about 10 minutes.

TRS-80 is a registered trade mark of Tandy Corporation.

- ☐ **16K Kit with shunts** (for computer) $ 95
- ☐ **16K Kit** (for expansion interface) $ 95
- ☐ **32K Kit** (for expansion interface) $180
- ☐ **48K Kit** (for computer and expansion interface) $265

California residents please add appropriate sales tax

Name (print) ______________________

Street ______________________

City ______________ State ________ Zip ________

☐ I've enclosed a check or money order for $ ______ payable to **UHF Associates.**

We honor: ☐ Master Charge ☐ VISA/BankAmericard

Account # ______________________

Expiration Date ______________

Signature ______________________

(required for charge card purchases)

UHF ASSOCIATES

90 Transport Avenue, #4
Rohnert Park, CA 94928
Call 707/584-7844

The Businessman's Business System

MSI Business Computer Systems offer flexibility and expandability unmatched by any other microcomputer system, large or small. Our SDOS operating system is totally device independent and supports up to seven users. This means that you can start with a single user, dual drive floppy disk system today, and add up to 80 megabytes of hard disk with additional workstations tomorrow. As your business grows, your MSI system grows with you—and your software won't become obsolete.

Perform text processing tasks at one workstation while entering sales orders on another. Add a third workstation in inventory control and a fourth in accounting. That's expandability!!!

- MSI Inventory Software, with complete Bills of Material, provides a complete inventory control and management system for manufacturers.
- Complete manufacturing forecasting, with production pick lists, allows automatic adjustment of component inventory levels.
- All transactions resulting in any change to the inventory data base are written to audit trail files listing date, time, operator's name, inventory item, and the changes which were made.
- Sales Order Entry/Accounts Receivable Software displays customer balances and credit standing as new orders are entered. Correct product prices and descriptions are obtained from inventory files if desired.
- Invoices are generated automatically as orders are shipped. Customer statements, with aged accounts receivable, are printed on demand.
- Purchase Order Entry/Accounts Payable Software optionally link to inventory program, in order to easily visualize inventory items which are on order.
- General Ledger programs link to the accounts receivable and accounts payable modules for easy updates and posting.
- If your business is expanding and you would like to know how an MSI Computer System can help you make it more profitable, call or write Midwest Scientific Instruments, 220 W. Cedar, Olathe, Kansas 66061, (913) 764-3273, TWX 910 749 6403 (MSI OLAT), TELEX 42525 (MSI A OLAT) ✓144

MSI ✓144 **Small Computers For Big Jobs**

Midwest Scientific Instruments